



# NEWS DIGEST

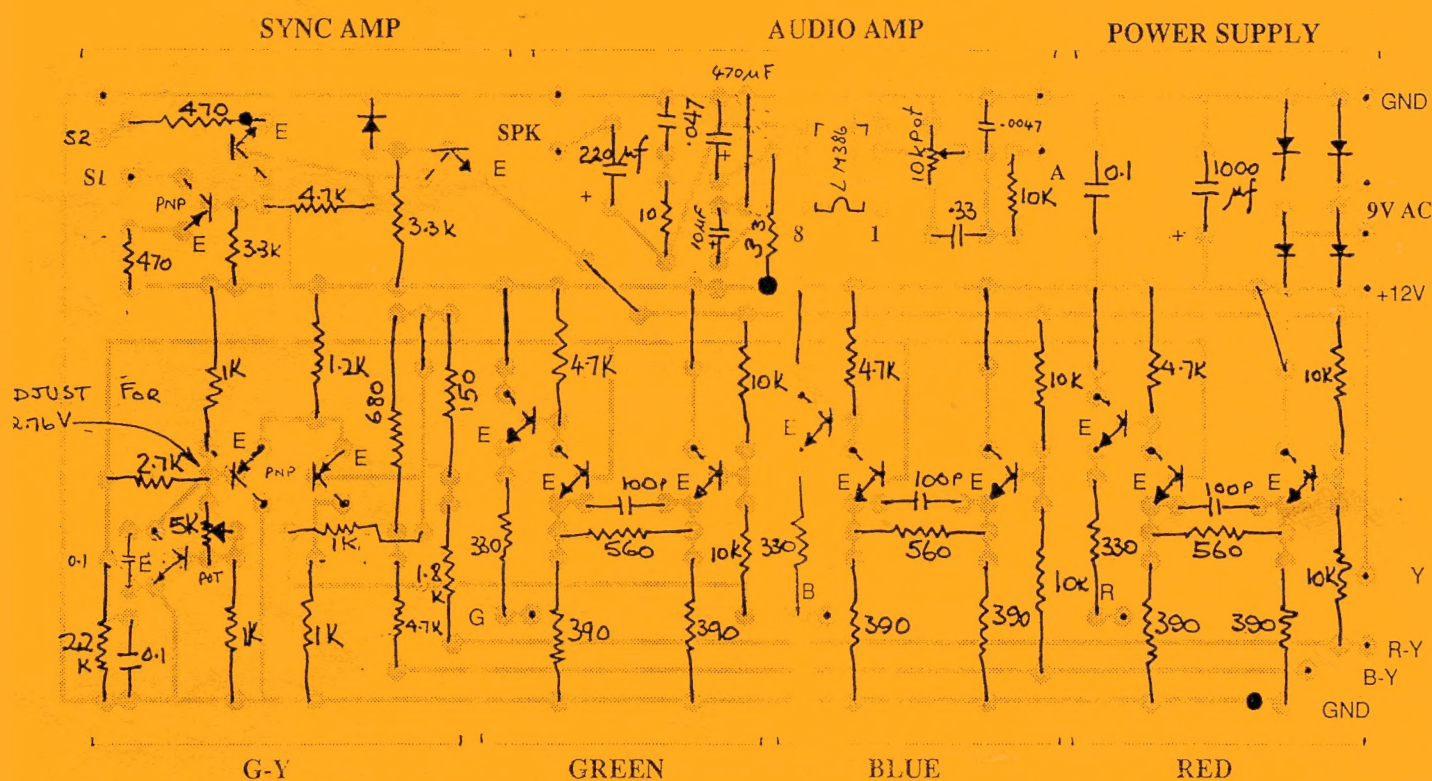
Focusing on the TI99/4A Home Computer

Volume 9, Number 1

January/February, 1990

Registered by Australia Post - Publication No. NBH5933

## Component overlay for interface PCB for Blaupunkt analogue RGB colour monitor





## TiSHUG News Digest

## Index

January and February 1990

All correspondence to:

P.O. Box 214  
Redfern, NSW 2016  
Australia

**The Board****Co-ordinator**

Dick Warburton (02) 918 8132

**Secretary**

Terry Phillips (02) 797 6313

**Treasurer**

Rolf Schreiber (042) 84 2980

**Directors**

Robert Peverill (02) 602 4168

Russell Welham (043) 92 4000

**Sub-committees****News Digest Editor**

Geoff Trott (042) 29 6629

**BBS Sysop**

Ross Mudie (02) 456 2122

BBS telephone number (02) 319 1009

**Merchandising**

Steven Carr (02) 608 3564

**Publications Library**

Warren Welham (043) 92 4000

**Software library**

Terry Phillips (02) 797 6313

**Technical co-ordinator**

Lou Amadio (042) 28 4906

**Regional Group Contacts****Carlingford**

Chris Buttner (02) 871 7753

**Central Coast**

Russell Welham (043) 92 4000

**Coffs Harbour**

Kevin Cox (066) 53 2649

**Glebe**

Mike Slattery (02) 692 0559

**Illawarra**

Geoff Trott (042) 29 6629

**Liverpool**

Larry Saunders (02) 644 7377

**Northern Suburbs**

Dennis Norman (02) 452 3920

**Sutherland**

Peter Young (02) 528 8775

**Membership and Subscriptions**

Annual Family Dues \$25.00

Overseas Airmail Dues AU\$50.00

**TiSHUG Sydney Meeting**

The next meeting will start at 1.30 pm  
on 3rd of February at Woodstock  
Community Centre, Church Street,  
Burwood.

Printed by

The University of Wollongong  
Printery

Title	Description	Author	Page No
Agenda for 1990 AGM	Club news	Phillips, Terry	2
Author index 1989	General interest		15
Brain, a review	Software review	Rosenberg, Howie	10
Co-ordinators report	General news	Warburton, Dick	2
Colour Monitor interface	Hardware project	Amadio, Lou	5
Colour Monitor interface	Hardware project	Trott, Geoff	5
Customizing Superbug 2	Software hints	Dohmann, Edgar	8
Extended BASIC Tutorial	Software hints	McGovern, Tony	9
Forth to you too!	Software hints		24
From the bulletin board	Mail to all		24
Getting a line on your program	Software hints	Presser, Joe	25
Minutes of 1989 AGM	Club news	Phillips, Terry	3
Page Pro 99	Software review	Johnson, Ed	27
Periscope instructions	Game	Pewterware	11
Regional group reports	General interest		31
SAVE/RECALL user notes	Software hints	Kaplan, Todd	23
Secretary's notebook	Club news	Phillips, Terry	4
Subject index 1989	General interest		14
Superbug 2, a description	Software review	Dohmann, Edgar	8
Techo time	Hardware project	Amadio, Lou	5
They're off	General interest	Trott, Geoff	1
TI-Base 2.0 review	Software review	Lilley, Rick	17
TI-Base tutorial	Data base	Smoley, Martin	13
TI-Writer version 4.0	Word processing	Green, Art	21
Tips from the tigercub #38	Software hints	Peterson, Jim	19
TiSHUG software column	Club software	Phillips, Terry	2

**Stop Press**

2 way interface PCBs now available from the shop at \$23 each  
Monitor interface PCBs now available from the shop at \$12 each  
Blaupunkt RGB analogue colour monitors (made for WANG), last shipment,  
now taking orders. They will cost \$122 each, including cost of interface PCB.  
The money must accompany orders. Contact Rolf on (042)84 2980

**They're off** by Geoff Trott

I hope you all had a Merry Christmas and a happy New Year. I guess our friends in Newcastle have been through a difficult time with the earth-quake just before Christmas. I hope that none of the Hunter Valley Group were too involved with that. I happened to be downstairs at the time contemplating some repairs or something when my wife and daughter came down to see what I had done to make the house shake! Innocent as usual, they left convinced that we had had a tremor, but it was not until I heard on the radio about the major problems in the Newcastle region that I realized something major had happened. I did not feel a thing! I slept through the other "major" earth-quake of my life when I was a lad in Adelaide too! That one led to quite large cracks appearing in our house and in quite a large percentage of the houses in Adelaide. I was actually in the middle of planning some structural changes downstairs as it happens, and I have installed the frame for a partition next to my computing area as the beginning of making the area more habitable. Having someone else to do the editing for me this month has meant some time for other things as well as a complete break from the console for a few weeks. This will be my last "They're Off" for a while at least but I am planning to write a series about how the computer works, which I hope everyone will be able to understand.

continued on page 30

# Co-ordinator's Report

by Dick Warburton

1990 promises to be an exciting time for TISHUG members. A series of interesting projects has been developed by our Technical Coordinator, Lou Amadio, which will interest a fairly wide range of members. Kits will be made available through the shop at the best possible prices. PCBs are already being made, and others are planned. As well I want to establish as soon as possible, a project group, which will allow members to help each other with the different activities, and perhaps help in developing new concepts for the TI.

## PROJECTS

1. 32K console memory expansion - \$30.
2. AT Multifunction Card..about \$130.
3. The I.O. Interface..and Stand-Alone System.eg. Disk Controller Card + ramdisk. Cost depends on extras.
4. Ramdisks..8K or 32K versions.
5. Ramdisks with basic software in eprom.
6. Console Multifunction Boards.
7. Speeding up the console.
8. RAMBO...A system to allocate ramdisk memory for program use.

## OTHER IDEAS

- A. A Gram card.
- B. An Alarm System.
- C. A Music Card.
- D. Monitor Interfaces.
- E. An I.O. Controller.
- F. An 80 Track Disk Controller.
- G. A 1.4 Meg Disk Controller.
- H. A Hard Disk Controller.

There must be a great number of interesting projects for us to try. If you have a suggestion, let me know and we will try to evaluate its feasibility.

I am also hopeful that in 1990, the club will purchase some up to date software for evaluation and demonstration. Again if you have come across really interesting software, let know so that we can consider its purchase. We also intend to keep up to date with the latest freeware.. As I said before, 1990 should be a good year.

Dick Warburton.

## For Sale

Peripheral Expansion Box	\$250
Double Sided Double Density disk drive	\$75
TI disk controller	\$75
or all three for	\$350
CTI CPB80 dot matrix printer (near new)	\$200

Contact Paul Barton on (02)603 5776

# Agenda for Annual General Meeting, 1990

TISHUG (AUSTRALIA) LIMITED - ANNUAL GENERAL MEETING 1990  
- WOODSTOCK COMMUNITY CENTRE, CHURCH STREET BURWOOD -  
TO BE HELD 3 FEBRUARY, 1990 COMMENCING 2.00PM.

## AGENDA:

1. OPENING
2. MEMBERS PRESENT AND APOLOGIES
3. READING AND CONFIRMATION OF MINUTES OF 1989 AGM
4. CORRESPONDENCE AND DEALING WITH SAME
5. READING AND DEALING WITH RECOMMENDATIONS FROM THE BOARD OF DIRECTORS - LIFE MEMBERS AND OTHER MATTERS
6. DIRECTORS REPORTS, PRESENTATION OF ACCOUNTS AND AUDITORS REPORT
7. UNFINISHED BUSINESS FROM LAST AGM (IF ANY)
8. ELECTION OF RETURNING OFFICER AND TWO (2) SCRUTINEERS
9. ELECTION OF DIRECTORS
10. ELECTION OF AUDITOR
11. NEW BUSINESS (IF ANY)
12. MEETING CLOSURE

IMPORTANT - members are requested to be in attendance by 1.30pm to enable them to sign in and ensure a 2.00pm start.

## TISHUG Software Column by Terry Phillips

At the December meeting there was a sell out of TI-ARTIST PLUS. Some new copies are being obtained so if you want a copy please let me know. Cost will be \$25 for new buyers or \$15 for members upgrading from an older version.

During 1990, the cost of disk software will be reduced from the present \$5 to \$2. The chief reason for the price reduction is the now lower unit costs of blank disks. The \$5 fee was set some years ago when disks were at a much higher price than they are now. It is also intended that much more publicity will be given to disk software available through the shop as well as imported items that members would like to see on sale. Thoughts on this latter point would be very welcome.

During 1989 there appeared to be a slackening off in disk receipts compared to previous years. Whether this is because of a slowing down in software releases for the TI and more emphasis being placed on the Geneve in the USA or some other reason I am not sure, but regardless this group is fortunate in having a comprehensive disk library for members to choose from. Few members however seem to take up the offer of selecting what they want from the library and having it available at say the next meeting. Provided you supply the disks remember this costs you nothing as I do not mind copying disks if members want them. It has been said in the past that members may not even be aware of what is in the software library. This problem is easily overcome. With 4 SSDD disks, 2 DSDD or 1 DSDD disk you can have a copy of the entire contents of the library. Please do not ask for a hard copy unless it is absolutely essential you have it in this format. Any member wanting the disk version only has to ask. Again if you supply the disks it is yours for nought.

continued on page 7

# Minutes of AGM, 1989

by Terry Phillips, Secretary

TiSHUG (AUSTRALIA) LIMITED - MINUTES OF ANNUAL GENERAL MEETING HELD AT BURWOOD RSL CLUB LIMITED, 96 SHAFTSBURY ROAD, BURWOOD NSW ON 4 FEBRUARY, 1989.

## 1. OPENING:

Chairman Cris Buttner opened the meeting at 1.40pm welcoming those members present. Members present shown on attached list with names highlighted.

## 2. APOLOGIES:

The following apologies were recorded and accepted:

Peter Schelling  
Eric Ockenden  
George Meldrum  
John Robinson  
John Scott

## 3. READING AND CONFIRMATION OF MINUTES OF 1988 ANNUAL GENERAL MEETING:

These minutes had been printed in a special January edition of the TND and circulated by post to all members. They were confirmed on the motion of Steven Carr, seconded Robert Peverill with the following amendment as proposed by Geoff Trott, seconded Rolf Schreiber:

AMENDMENT: Delete from Item 11, New Business, reference to Publications Library - Warren Welham.

## 4. CORRESPONDENCE AND DEALING WITH SAME:

There was no correspondence for discussion.

## 5. READING AND DEALING WITH RECOMMENDATIONS FROM THE BOARD - LIFE MEMBERS AND OTHER MATTERS:

At this juncture, Chris Buttner presented a certificate of appreciation to Ross Mudie for his work on behalf of members on the BBS and other club activities. A second certificate was to be presented to Vincent Maker for his contribution to the TND and his attempts in fostering interest with younger members. As Vincent was absent his certificate will be mailed.

Chris Buttner advised that the Board had no nominations for honorary life members, but queried whether there were any from the floor. Rolf Schreiber proposed Ross Mudie and Geoff Trott. Both nominations were seconded by Steven Carr. Both nominations were approved on a show of hands by the members present.

## 6. DIRECTORS REPORTS, PRESENTATION OF ACCOUNTS AND AUDITORS REPORT:

Chris Buttner advised members that they should amend their copies of the report to show membership as 236 in lieu of 336 under the headings Membership and Note 1. He also advised that the graph provided with the reports was inaccurate and correctly should show the shop trading in a slight profit situation. Amended graphs were passed around for members perusal.

Chris then invited questions from the floor on the reports:

Geoff Trott - queried income tax liability and on what the group was liable for income tax. Chris Buttner explained that the group was only liable to pay income tax on interest from investments plus sales to non members, although there had been no such sales in the past 12 months. Tax bills were for the previous 3 years assessments and only paid in the past year.

Geoff Trott - queried Administration Income/Expenses. Percy Harrison explained that Administration income comprised fees paid for BBQs and entrance fees to

tutorials and interest received on investments. Capital assets were debited to Administration as were meeting fees and miscellaneous postage, but not postage associated with the TND.

Rolf Schreiber - queried amount expended on Auditors fees. Percy Harrison explained that cost based on hours worked by auditor in preparing accounts. It also included a component for preparation of tax returns. Percy Harrison further mentioned that an account had not yet been received from the auditor for the preparation of the 1988 accounts.

There being no further questions the accounts were accepted as read on show of hands after a motion by Alf Culloden, seconded Derek Wilkinson.

## 7. UNFINISHED BUSINESS FROM LAST AGM (IF ANY):

Nil.

## 8. ELECTION OF RETURNING OFFICER AND TWO (2) SCRUTINEERS:

Ross Mudie elected as returning officer on the nomination of Robert Peverill, seconded Alf Culloden. Geoff Trott elected as one scrutineer on nomination of Rolf Schreiber, seconded Lou Amadio, John Paine elected as second scrutineer on nomination of John O'Brien, seconded Bert Thomas.

## 9. ELECTION OF DIRECTORS:

The retiring Directors vacated the chair and Ross Mudie took over the meeting as Chairman. Ross moved a vote of thanks to the retiring Directors, and this was done by acclamation from the members present.

Ross advised that only 3 nominations, Terry Phillips, Richard Warburton and Russell Welham, had been received and called for further nominations.

Robert Peverill accepted his nomination by Lou Amadio, seconded Chris Potts.

Rolf Schreiber accepted his nomination by Craig Sheehan, seconded Bert Thomas.

There being no further nominations the Returning Officer declared Messrs. Peverill, Phillips, Schreiber, Warburton and Welham elected.

## 10 ELECTION OF AUDITOR:

Rolf Schreiber nominated David McNiece, and this nomination was seconded by Lou Amadio. Chris Buttner explained that there were certain requirements in appointing an Auditor and that it was important that the person nominated could audit the books over the Christmas period. A second nomination was Graeme Smith, nominated Percy Harrison, seconded Chris Buttner.

On a motion from David Andrews, seconded John Paine the meeting agreed to evaluate both before making a decision to appoint one of the nominees.

## 11. NEW BUSINESS (IF ANY):

At this point Chris Buttner asked the new Directors who was who. Richard Warburton advised the meeting of the following:

Chairman - Richard Warburton  
Secretary - Terry Phillips  
Treasurer - Rolf Schreiber  
Director - Robert Peverill  
Director - Russell Welham

Les Andrews queried the new Directors on their computing interests.

Richard Warburton advised he was a sole TI user and had no intention of diversifying.

Russell Welham advised he had diversified but still rated his TI number 1.

David Andrews queried where the group was heading.

Rolf Schreiber felt there would be some changes down the track if the group was to hang onto its existing membership base and encourage usage of the machine. He further felt that the strength of the group lies in Regional Groups and contact that can be made at that level.

Robert Peverill suggested that all members should encourage people that they know own a TI to join the group.

Les Andrews would like to see the formation of special interest groups to cater for members various interests.

Chairman Richard Warburton moved to the election of the various sub-committees:

PUBLICATIONS LIBRARY - Warren Welham volunteered.

BBS/SYSOP - Ross Mudie volunteered.

TND - Some discussion ensued. Ross Mudie advised that he would like to see the production of the TND by the Wollongong group continue. Geoff Trott advised he would like to continue but would like more support. He saw it as essential he be supplied with more equipment to cut down production time. Minimum requirements included Press, 80 column support and PE Box. Geoff felt he had received little support from Directors during his term as Editor. Chris Buttner disputed this and advised that Directors had made offer of fully expanded system to Geoff on proviso that it may sometimes have to be made available at meetings. Geoff had rejected this offer preferring a mini-system. Chris further pointed out to the members the large expense of providing Geoff with what he wanted to carry on. Chris Potts' feelings were that all expenses need to be justified and not at the detriment of other group activities. Ben Von Takach advised that it would cost approximately \$1000 to supply Geoff with what he viewed as necessary. John Vanderney offered Geoff the use of a 30 meg drive and HD controller. John Paine offered the use of DSDD disk drives. Richard Warburton asked that the matter be left with the Directors who would consider Geoff's requests sympathetically.

SHOP - Steven Carr volunteered but would like assistance from a willing band of assistants.

MEETINGS - Craig Sheehan and David Sullivan volunteered to plan meeting agendas.

PUBLICITY - No Volunteers.

SOFTWARE - Terry Phillips volunteered.

TECHNICAL - Lou Amadio volunteered and John Paine advised he would assist where he could.

Geoff Trott felt that both the Publications Library and BBS were generating insignificant revenue and that all members should be encouraged to use all facilities. He urged the Board to consider dropping fees for these services in an effort to encourage greater usage.

## 12. MEETING CLOSURE:

As there was no further business for discussion, Chairman Richard Warburton closed the meeting at 3.57pm thanking all for their attendance.

Minutes Recorded By:

Terry Phillips  
Honorary Secretary

4 February, 1989.

Minutes Confirmed By:

Dick Warburton  
Chairman

o

# Secretary's Notebook

by Terry Phillips

Well what a great year we had in the group, and it was really capped by a great fine day for our end of year BBQ back on December the 2nd. Boy that seems a long time ago. I trust that all members and their families have had a good festive season and that this year, 1990, is both healthy and prosperous.

Before going on to review what happened during 1989 there are four new members to give a big welcome to. And they are:

Chris Parris - Cowra NSW  
Alan Comarmond - Carlingford NSW  
Remo DeGiovanni - Lansvale NSW  
Robert Kelly - Babinda QLD

And speaking of membership, here is the current situation.

At the end of 1988 we had 239 members (including 3 honorary life members). During the year, 1989, 32 new members joined the group, however 62 elected not to continue with their memberships. The result is that at the end of 1989 there are 209 members (including 5 honorary life members). My prognosis for the year ahead is that most of the current membership will continue and that the group will survive into the years ahead with a membership base of around the 200 mark. The Bulletin Board, by the way, has 84 members, so a good percentage appear to be into modem communications.

Exchange newsdigests are provided to all Australian User Groups as well as 14 overseas groups. Unfortunately some local and overseas groups appear to be falling on hard times, and resulting frequency of issue of their newsdigests is down. As the cost of production rises this is an area which will have to be closely looked at during 1990.

Complimentary copies of our newsdigest are also forwarded to Tony McGovern, Jim Peterson and MICROpendium in recognition of their whole hearted support to the TI Community. In addition copies for archives are forwarded to the NSW State Library (Sydney) and the Australian National Library (Canberra).

Your Board met 9 times during 1989 and attendance recorded was as follows:

Dick Warburton (Chairman) - 9  
Terry Phillips (Secretary) - 9  
Rolf Schrieber (Treasurer) - 7  
Robert Peverill (Director) - 7  
Russell Welham (Director) - 7

During 1989 there were 10 general meetings with a wide range of topics discussed and/or items demonstrated. Thanks should go to Craig Sheehan in particular for his untiring work throughout the year in coming up with new ideas for these meetings. Two full day workshops were held, the first in June and the second in November. The November event marked the first time we had been away from Woodstock, apart from AGM's, for a considerable period. This event was held at the Ryde Infants School and this venue looks a very nice setting for future full day events. We were plagued with a few problems at Woodstock and we could not always get the number or rooms we needed. One, however, needs to look at this in perspective and consider the huge number of community groups that utilise Woodstock facilities. Most times at Woodstock we can be guaranteed the large upstairs room and possibly one other room.

The event that stands out in my mind the most during 1989 was the August meeting with its buy, swap and sell theme. I would say that it attracted the largest gathering throughout the year with most members reporting they had picked up a bargain or two, or had sold what they brought along. Perhaps a couple of meetings on these lines may be the go during 1990.

continued on page 7

# Techo Time

with Lou Amadio

The club Christmas get together last year was my first, and it was a most enjoyable event. For some reason I thought that this was going to be a fairly casual meeting. But things were as hectic as ever.

## RGB Colour Monitors

We are currently purchasing 20 RGB analogue monitors for club members. All 20 are sold but I am confident of being able to get some more. These are available for \$110 each but we have to buy them in lots of 10. So far I have interest from 4 club members. Anyone wanting a monitor should contact me (Lou) as soon as possible on 042-284906. It will take about 1 month from time of order to delivery.

By the time that you read this, Geoff Trott will have the monitor interface design finished and a printed circuit designed. Members are encouraged to build their own interface as this will work out a lot cheaper. Geoff is trying to design an interface which will require the minimum of adjustment.

## More on The Hard Disk Power Supply

Readers of this column will know that one of my aims in designing and/or using electronics circuits is to keep heat dissipation to a minimum. However, because of the wide variation in current requirements of some circuits, this is not always possible. Let me explain

Having just purchased a Hard and Floppy Disk Controller, courtesy of Ben Takach, I set about building a power supply for my full height hard disk. Preliminary measurements had indicated that the supply would need to deliver about 3.5 amperes at start up falling to about 1.4 amperes when the disk was up to full speed. Naturally I decided to use the power supply design that was published in the Sept '89 TND. All that was needed to adapt this design to the higher current requirements of the full height was a larger transformer and a larger heat sink.

The transformer that I elected to use (supplied by Rolf) had two separate 15 volt windings rated at 2 amperes. Each winding was tapped at 12 volts with a slightly higher rating of 2.5 amperes. When used in parallel, the 12 and 15 volt AC windings could deliver 5 and 4 amperes respectively.

Using the circuit design published in the Sept '89 TND, the 12 volt windings in parallel could not deliver sufficient current during the critical start up phase (Geoff estimated that the full height drive needed about 5 amperes DC for the first few seconds of operation). Consequently, I had to revert to using the 15 volt AC winding. The drawback, however, was the much higher heat dissipation in the series pass transistor requiring a much larger heat sink.

On the way back from the December meeting, it occurred to me that I could have my cake and eat it. I could use the 15 volt AC windings to start the hard disk and then then switch to the 12 volt AC windings when the hard disk is up to speed. A quick solder job with some wires and a switch showed that the solution worked perfectly.

## Single Chip 32K Memory Expansion

Due to the need to produce documentation for the RGB monitor interface, the second article on memory expansion has been held over until next month.

### Multi-Function Cards

Circuit boards, components and building instructions are expected to be available from the club

shop in February. It is estimated that a double density disk controller configuration would cost about \$130, while a fully configured board which includes 32K, PIO and RS232 and clock is expected to cost about \$200. This is a very economical way to upgrade your TI99/4A system.

## Colour Monitor Interface

by Lou Amadio and Geoff Trott

Note: This article is being written prior to the production and testing of the printed circuit version of the interface. PCBs should be available through the club shop by the first meeting in 1990. Two prototypes of the circuit have been built on veroboard and are currently being used by Geoff and myself.

## Colour Monitors

New member Bob Keast has supplied an initial order of 20 RGB monitors for TishUG members. These arrived just before Christmas. The monitors are ex work station high quality 80 column units made for Wang Australia by Blaupunkt (Germany), and selling for \$110. A second order for ten has been placed and to date 7 have been reserved. Contact Rolf or myself if you are interested. According to Bob, these monitors are only available for a very limited time.

## Interface Design

Since the Peter Schubert RGB interface did not work with this monitor, Geoff set about designing a new one.

The current design, in conjunction with the Wang monitor, produces a very high resolution picture where the individual pixels can be readily seen.

For those not familiar with video signal processing (including myself), the following is my understanding of the subject from talking to Geoff:

The console Video Display Processor produces 3 signals: R-Y, B-Y and Y. The first 2 contain the colour information, whilst the Y signal contains both video and synchronizing signals. The primary colours (red, blue and green) are generated by suitably mixing these 3 signals:

$$(R-Y) + Y = R \text{ (red drive)}$$

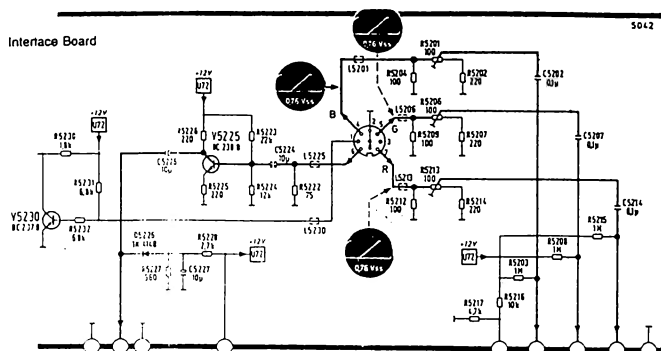
$$(B-Y) + Y = B \text{ (blue drive)}$$

$$0.591*(R-Y) + 0.186*(B-Y) = G-Y$$

$$(G-Y) + Y = G \text{ (green drive)}$$

The red and blue components have to be mixed in certain proportions (ratio = 2.7 : 1) in order to produce green. Note the unequal input resistors on the G-Y amplifier.

For this design, Geoff decided that the best way to generate green was to synthesize a G-Y signal from the R-Y and B-Y signals. In this way the final design was easier for the user with only one preset potentiometer.



## Sync Amplifier

According to Geoff, most of the difficulty with the design centered around the sync amplifier. At least 5 different designs were tried before he was happy with the outcome. He found that the quality of the screen display was very much dependent on the shape of the sync signal.

## Using Other Monitors

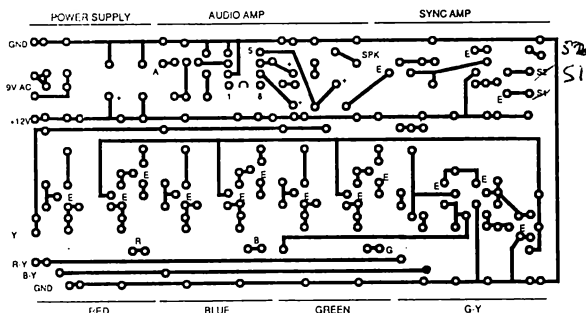
It should be possible to use this interface with IBM CGA colour monitors. This would require adjusting the 330 ohm output resistors and designing a sync separator circuit. The current design provides for both positive and negative sync outputs.

## Audio Amplification

For the PCB layout, I used the audio amplifier design published by Peter Schubert (refer TND May 1988). The parts list in this article does not include the components required for the audio amplifier.

## Interface Parts

The following list does not include the parts required for the audio amplifier (refer to the May 1988 issue of the TND).



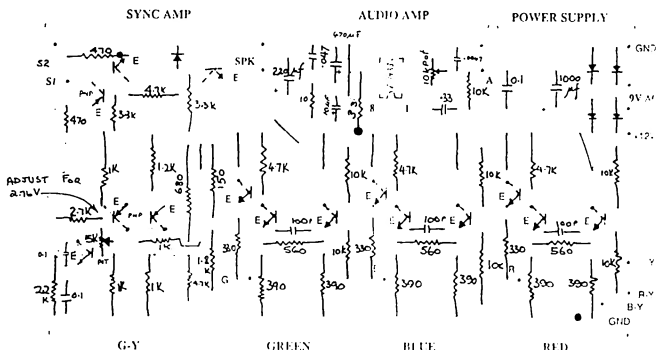
Interface PCB

Qty	Value	Comment
12	BC547	NPN Transistor
3	BC557	PNP Transistor
5	1N4001	power diode
1	1000 uF	filter capacitor
3	0.1 uF	capacitors
3	100 pF	capacitors
1	150 ohm	resistor
3	330 ohm	resistor
6	390 ohm	resistor
2	470 ohm	resistor
3	560 ohm	resistor
1	680 ohm	resistor
4	1K ohm	resistor
1	1.2K ohm	resistor
1	1.8K ohm	resistor
1	2.2K ohm	resistor
1	2.7K ohm	resistor
2	3.3K ohm	resistor
5	4.7K ohm	resistor
6	10K ohm	resistor
1	5K ohm	trim pot
1	50 mm	speaker
1	9 volt AC Plug Pack (or 12 volt DC Pack)	

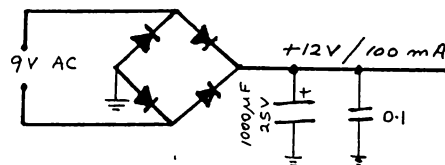
## Construction

Use the component overlay diagram to assist in building the interface. Most of the transistors are low cost BC547 (NPN) with a small number of BC557 (PNP).

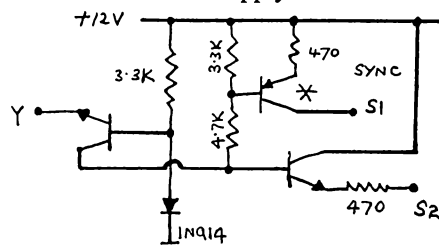
The 5.38K ohm resistor consists of 4.7K in series with 680 ohm. The 1.95K ohm resistor consists of 1.8K ohm in series with 150 ohm.



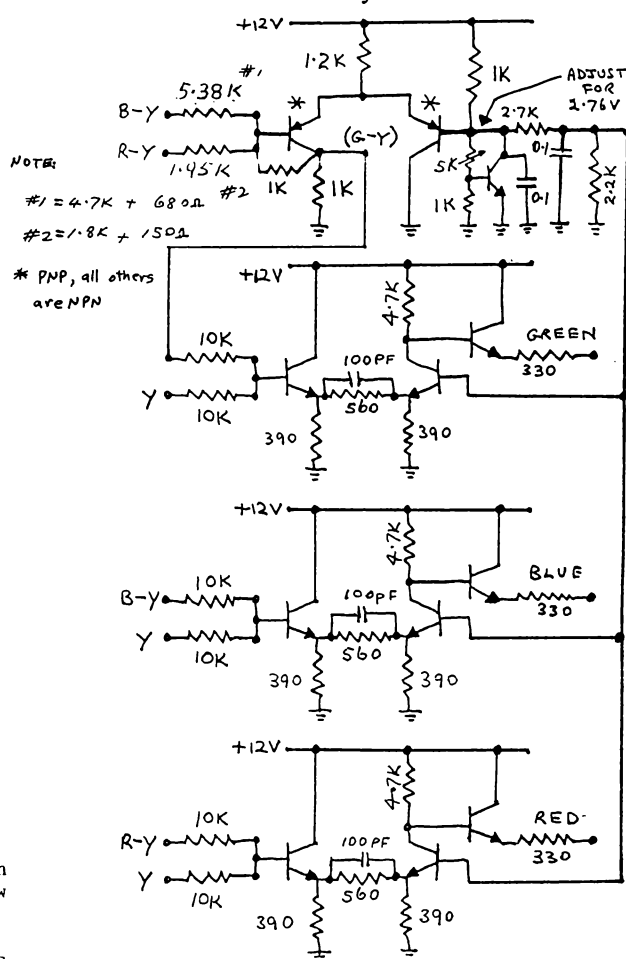
Component overlay for PCB



12 volt Power Supply for Interface



Vertical Synchronizing circuit



Interface circuit schematic diagram



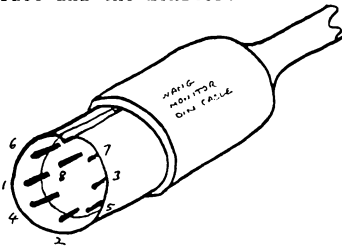
For the console to interface connection, use the cable from your modulator. Note the wire colours corresponding to the Y, R-Y, B-Y, Audio and ground connections from the modulator PCB.

The emitters of the transistors are marked on the PCB overlay. Note also the polarity of the electrolytic capacitors, diodes and the LM386 audio chip.

I recommend a Dick Smith Electronics Zippy Box (50 x 90 x 150, H2851) to house the PCB, 50 mm speaker and 10K volume control pot. A larger speaker would produce a better sound. The circuit is powered by a plug pack transformer (either 9 volts AC or 12 volts DC) and draws about 100 mA. Do not use the +12 volt supply from the console for this interface.

An 8 pin Din plug and cable should be supplied with the monitor. Use the following table and diagram to connect between the interface and the monitor:

Pin	Wire	Function
1	White	n/a
2	Braid	Ground
3	Yellow	n/a
4	Blue	Blue Drive
5	Green	Green Drive
6	Black	Sync Drive
7	Red	Red Drive
8	Braid	Ground



If the cable supplied with your monitor has different colours, use the monitor input diagram and an ohmmeter to trace the connections on the 8 pin Din plug and lead.

#### Interface Adjustment

Geoff made it very easy to set up the interface. Simply adjust the 5K ohm trim pot for 2.76 volts DC at the junction of the 1K ohm and the 2.7K ohm resistors.

#### Monitor Adjustments

Two different types of colour monitor were supplied by Bob:

Compact model #6525-2.

Swivel base model PM005.

Several internal monitor adjustments are required before it will work properly with the console. Take care that you do not change the setting on the wrong potentiometers, particularly the 115 volt adjustment.

1) Adjust the vertical frequency oscillator trim pot to lock the picture. This pot is located on the motherboard. For the compact model (#6525-2) this is pot R707 near IC V720. For the model with the swivel base (PM005) there is not enough adjustment to lock the picture. For this model, change the 10K resistor (R541) associated with the 5K trim pot to 15K ohm. Cut out the old resistor and mount the new on the back of the motherboard. The trim pot on the PM005 that I modified did not need any further adjustment.

2) Adjust the horizontal and the vertical picture size trim potentiometers (in that order) to suit your monitor. For the compact model, these are R942 on the equalizer board (perpendicular to the motherboard) and R704 on the motherboard respectively.

For the swivel base model, these are R946 and R965 both on the equalizer board. Picture horizontal position may be changed with R939.

In either case, these potentiometers should be adjusted to suit your needs. Remember that picture quality is inversely proportional to picture size.

3) Adjust the trim potentiometers behind the 8 pin Din input socket on the monitor for maximum gain on the R, G and B colours.

4) Carefully adjust, if necessary, the R,G and B drive trim potentiometers on the RGB board located at the topmost part of the chassis.

#### Monitor Circuit Diagram

Please note that circuit diagrams for Model #6525-2 will be available for perusal at the meetings. I am currently trying to get a copy of the service manual for the swivel base model.

#### continued from page 4

Finally, it would be remiss, not to thank Dick, Rolf, Robert and Russell for their help throughout the year with the management of group affairs. To Ross who untiringly runs the Bulletin Board, to Stephen who does pretty much a thankless job in the shop, and to Lou who seems to keep coming up with great technical ideas go my congratulations for their efforts during the year. And last, but by no means least, to Geoff who puts together the best TI magazine, I feel we all owe a personal thanks and congratulations. Without his contribution with the TND which keeps us amused, occupied and computer educated, I guess the group would not amount to much. Thanks Geoff.

See you at the AGM.

#### continued from page 2

It should also be mentioned that we are fortunate in having the entire collection of Jim Petersons commercial software range including his NUTS BOLTS disks. These are available at a very reasonable cost of \$5 per general disk and \$10 for NB. This range of software was publicised extensively in the May 1989 TND but I must say sales have been disappointing. We do not have all of Jim's public domain collection, however we do have the bulk of it.

Have you written something that you would like to donate to the library, or do you have something that maybe the library has not got. Why not consider donating?

#### continued from page 8

The default output device specification is near the front of the file. Start searching about 500 bytes or so into the program for the character string:

RS232.BA=9600

This string will be followed by 7 spaces so you can replace this string of 20 characters with your own default output specification. Be sure not to exceed 20 characters or you will overlay area that is used by the program which will mess up any name longer than 20 characters. Also be sure to change the byte immediately preceding the printer name from >OB to the length of your new device name.

The five colour codes are located near the end of the program. Start searching about 100 bytes from the end of the program for the following sequence of data:

F4F4  
1B1B  
1313  
1E1E  
4E4E

These are the 5 colour combinations which you toggle through with the <J> command. Change them to whatever combinations you wish to use. It is relatively easy to change these 5 codes to other values but it would take major surgery to add more colours to the list.

If you wish to add more choices, to the list, the code for toggling through the list immediately follows the list and if you dis-assemble it you may be able to figure out how it works and decide how you might like to change it. Again, be sure you have a backup copy of the original distribution before you try this.



# Superbug II

by Edgar Dohmann, USA

Thanks to a lot of help from Ewell Brigham of the Houston User's Group, version 2.0 of SUPERBUG II will soon be available. I am hoping for a May 1, 1986 release date if I can get the manual updated and printed by then.

Two major new features of ver 2.0 are the ability to load and save program files. The load feature will operate in a similar fashion to option 5 of the Editor/Assembler. However, if you want to load the program into a different area from its default location, this can be done with the new version of SUPERBUG II.

The save program file is similar to the SAVE utility on the disk supplied with Editor/Assembler. However, with SUPERBUG II, the SFIRST, SLAST, and SLOAD labels do not have to be DEFINED in the object file. You supply the starting and ending address when you activate the save function and it saves the program in the same way the SAVE utility does.

Other changes from version 1.0 are mainly cosmetic but a few remaining bugs are also fixed. The remaining changes include:

- 1) The J command is changed so the border colours are also changed when screen colours are changed.
- 2) A bug in the M and D commands is fixed. This bug only showed up when memory dumps crossed address >8000.
- 3) A bug in the D command to an external device is fixed. This bug only showed up when VDP or GROM memory was dumped to an external device.
- 4) The Q and E commands have been improved, especially for the SUPER SPACE version of the program.
- 5) The small character sets are automatically loaded when the program is started up from Console BASIC or from SUPER SPACE.
- 6) The Console BASIC startup now works like the Extended BASIC startup. In version 1.0, the initial prompts were not visible from Console BASIC.
- 7) The leading zero is removed from registers R0 through R9 in the disassembler. This allows the code produced to be easily reassembled.
- 8) A bug in disassembly of JMP instructions is fixed so the operand value will reassemble properly.
- 9) A modification for writing to GRAM is added for compatibility with GRAM KRACKER.

There are still a few additional features I would like to add to the program but it is a full 8K in size at the present time and that is all that will fit into SUPER SPACE. I am trying to do a little more code squeezing and if I can save enough space, there may be a few more goodies added. Any such additional features may have to wait for version 3.0 however.

The distribution disk for version 2.0 will not have a complete manual on the disk as with version 1.0. For one thing, with the additional information to describe new features, the complete manual will not fit. In the second place, I feel that the manual-on-a-disk feature was an experiment that did not work out well.

When I distributed SUPERBUG II through public domain FAIRWARE, I requested a contribution of \$3.00 to \$5.00 from anyone who got a copy in this manner. I also promised a printed copy of the manual to anyone who sent at least \$5.00. So far I have only received ONE contribution in this manner. I know that a large number of User's Groups have a copy of my program in their

library and it is hard to believe that only one person has obtained a copy in this manner. I assume that because the disk was totally self-sufficient that most people do not see the need to send a contribution and get a printed copy of the manual.

On the other hand, I have received over 150 orders from publicity generated by MICROpendium. I have also sold about 20 copies through authorized distributors at various /4A fairs. Only because of this response have I had the initiative to complete version 2.0. Since those people who order a copy from me directly also get a printed copy of the manual, there is no real need for the manual on the disk.

The new disk will have a short help file that will explain how to load the programs and will offer a printed copy of the manual to anyone who sends me a contribution of \$5.00 or more. People who order copies from me directly will continue to receive a printed copy of the manual.

I will be continuing the same ordering policy that was in effect for version 1.0. Anyone may receive a copy of the program with a manual if they send me \$10.00 or if they send me a disk, mailer, return postage, and \$5.00.

Anyone who already has version 1.0 and wants to upgrade to version 2.0 will have to follow the same procedure used to obtain a copy the first time. Since the price I charge is so minimal, I cannot afford any other type of upgrade service.

I sincerely appreciate the orders I have received and the ONE public domain contribution. Many people have followed up their initial correspondence with more letters. I really enjoy the opportunity to get to know other users around the country better and appreciate all the suggestions and comments that were sent. I hope that version 2.0 of SUPERBUG II will also be well accepted.

## Customizing Superbug II

by Edgar Dohmann, USA

I appreciate all the interest that has been shown in SUPERBUG II. I spent a lot of time working on this program and the manual for it and it is quite rewarding when people write or call me to comment on the program or make suggestions for improvement. It is also nice to receive a few dollars from people who think enough of the program to help compensate for some of my time and effort. Thanks to all of you and also thanks to THE NATIONAL NINETY NINER and MICROpendium for carrying articles and announcements about SUPERBUG II.

The purpose of this article is to explain how you can customize a couple of SUPERBUG II features to suit your own needs or desires. I had intended to include this information in the manual that is on the disk but I used up all available sectors before I got this far. The most likely portions of SUPERBUG II which you might like to customize are the default output device specification and the screen colours which you can toggle with the J command. You will need a Disk Patch program like the Disk Fixer, Advanced Diagnostics, Disk + Aid, or Ed's Disk Analyzer to perform these modifications. Be sure to make a backup copy of the SUPERBUG II distribution disk you have received before attempting these conversions. This serves two purposes: you can thus preserve a copy of the original authorized distribution disk to pass on to others, and second you will have a backup in case something goes wrong in the disk patching process.

Locate the file you wish to patch on the disk: SBUG, SBUG6, or SBUGO. SBUG and SBUG6 are in memory image (Program) format while SBUGO is in compressed object format. As a result, the areas you will want to patch will be different for each file. Hopefully, your disk patch program has hints on how to search, interpret, and patch each of these types of files.

continued on page 7

# Extended BASIC Tutorial

by Tony McGovern, Funnelweb Farm

## VII. ACCEPT AT and other RAMBLINGS

TI Extended Basic is a very substantial language. The XB cartridge contains 12K of ROM and 3 and a bit (the 4th one is not full) GROMs at 6K apiece. This is on top of the 8K of console ROM and whatever parts of the 3 console GROMs are still used in XB. The tragedy of the TI-99 is that GROMS and GPL were ever invented. I guess it was TI's way of trying to keep the software market sewn up. The end result as we all know is that they shot themselves in both feet with uncanny accuracy. Instead of using the TMS9900 CRU addressing to bank switch plain ordinary ROMs or even just using GROMs only as sources of code to load into RAM (as I believe is done in the p-code card), they could have had a machine that did justice to its CPU, a real home minicomputer ..... that is all past history now.

I have been pondering on what TI should have done way back when the 99/4 was first designed, that could have been easily done at the time (or even when it was updated to the 99/4a). My conclusion is that the machine should have been given 4K of fast 16-bit CPU RAM instead of a measly 256 bytes. There would have been plenty of room with a little rearrangement and/or better decoding of memory-mapped devices (VDP, sound, speech, GROMs). This would have meant that Basic and XB system areas, sprite tables, full screen buffers, string buffers, value stack, and so on could have been in fast RAM, and even console Basic could have had full scope for character and sprite definitions (as in TI-LOGO for instance). Their cartridges could then have easily been a lot better, and let's face it, many of the earlier ones were pretty hopeless, and the later ones are all limited by lack of honest CPU RAM. The only cartridges which have stood the test of time are those that use the 32K RAM expansion. TI would then have never been dragged into that marketing war to the death (TI's that was) with that vastly inferior machine, the VIC-20. I have a suspicion that the 256 bytes happened because part of TI management wanted to protect their existing evaluation board and smaller minicomputer business.

The immediate improvement really needed in XB sub-programs is a means of examining variable values in any sub-program when program execution is halted by BREAK or errors. TI should have done it in XB by retaining the EDIT command of console Basic, allowing it to access user subprograms by name. Anyone listening out there? If so add single command array operations, full syntax checking on entry, 80 column display capability with formatting power to match, bit-map screen functions, fast program execution and anything else will then be gravy. Then TI-99/4a owners will be most pleased to join in. The bad news is that TI is starting to cut back on support for the 9900 family despite its excellent qualities, and so it is becoming less attractive for new designs. In retrospect we still do not have these things in Basic, as the Geneve Basic remains incomplete and buggy (mid-89).

Enough ramblings and back to the tutorials ! What then is the most powerful feature in XB after SUB and CALL? A good candidate is the file system, but as this is already built into the console I will stick with commands specific to XB. The prime candidate is ACCEPT AT and its qualifying clauses (even just plain ACCEPT has some interesting improvements over INPUT but that has been treated elsewhere). This was emphasized by the recent appearance (mid-84) in a computer magazine of a long article on machine code for adding this function to IBM PC Basic (which does not have sub-programs either). ACCEPT AT is very useful and powerful, but has some undocumented features as well as some subtle and treacherous bugs, and is well worth talking about in this series.

The simplest level of ACCEPT AT combines the INPUT routine with its access to editing features, with cursor

positioning on the display screen by the AT clause. So far this is just the input version of DISPLAY AT. The difference from INPUT is that there is no provision for prompt strings, but a DISPLAY AT soon fixes that. It also accepts input to a single variable only, and not to a whole variable list. As ACCEPT AT and DISPLAY AT do not scroll the screen, their repeated use can give a much better effect than INPUT when graphics elegance is important. Construct your own examples here or work the XB manual examples. Remember that the cursor is in XB color group 0 if you are trying to dress up the graphics.

BEEP allows an audible prompt with only one program byte (we will talk about program length later on if this series keeps going long enough). Of course constant repetition of beeps can get a little wearing. The ERASE ALL clause provides an alternative to CALL CLEAR for clearing the screen. As compared with CALL CLEAR, ERASE ALL is slower to execute, (it seems to be line at a time) but takes less program space. Its effect is slightly different also. This little program which uses ERASE ALL with DISPLAY will make both speed and screen effects easy to see.

```
100 CALL CLEAR :: CALL COLOR(0,3,3)
110 FOR I=1 TO 100 :: CALL CLEAR :: NEXT I
120 FOR I=1 TO 100 :: DISPLAY ERASE ALL ::NEXT I
130 CALL SCREEN(11):: FOR I=1 TO 1000 :: NEXT I
```

Those are the simple pieces of ACCEPT AT -- now it starts to get interesting. VALIDATE allows the programmer to decide what characters are acceptable in a response. The computer honks (that is the word in TI-FORTH) at unacceptable inputs. Three predefined types are available. UALPHA accepts only upper-case alphabetic characters -- very useful for filenames and suchlike. This is not quite the same as depressing the alpha-lock key as it only accepts letters, and so is incompatible with input to a numeric variable. If you are in the habit of verifying wet paint signs by touch, try that for a change. The DIGIT type does just what its name implies, and NUMERIC allows the input of any floating point number as well as plain positive integers. As with INPUT, all numbers are acceptable to a string variable, but numeric variables are fussier.

Now what if these predefined types are not right for what you want ? Suppose only digits 1 to 4 are acceptable, as in a menu choice of 4 items labelled 1 to 4. In console Basic extra lines of code would be needed to check the input, but ACCEPT AT handles this with the clause VALIDATE("1234") or VALIDATE (I LIKE IT\$) where the string variable has previously been set to "1234". To put it more formally, only the characters in the string argument of VALIDATE can be entered at the keyboard to be ACCEPTed.

The SIZE clause allows ACCEPT AT to be used with almost no interference to screen displays. It blanks out the specified number of characters, providing an input window of finite length, and if the length specified is negative, the characters already in the window are not erased, and form an immediate input for ACCEPTance. This is very handy for making default choices obvious to the user. Let's enter a little program to get at the essentials.

```
100 CALL CLEAR :: DISPLAY AT(12,1):RPT$(" ",28)
200 ACCEPT AT(12,2)SIZE(3):A$
300 DISPLAY AT(15,2):A$;LEN(A$)
400 CALL KEY(0,K,S) :: IF S>0 THEN 100 ELSE 400
```

You most likely have the Alpha-lock depressed. If so let it off, and RUN our little program. Just press ENTER the first time round, next time hit <space> first, and finally <space> first before hitting another key. This shows that <space>s after the last honest character entered are ignored. Try some VALIDATEs here too, if you wish. Now with the program as given, alter SIZE(3) to SIZE(-3). It now ACCEPTs whatever is in the was or is placed in that 3 character input window.

Now that is all very simple, but it brings us to the edge of the undocumented wilderness. Alter the CALL

KEY(0,K,S) in the last line to CALL KEY(3,K,S) and RUN the program again, this time entering letters. Observe what happens the second time around. This answers the question of what keyboard mapping ACCEPT AT uses -- like CALL KEY(0,K,S) it uses the last one, whatever that was. Try split keyboard units in the last line. At the machine code level, a particular byte in the CPU scratchpad RAM has to be set to the key unit before calling the SCAN routine. I interpret the behaviour as showing that in the XB modules of my experience that ACCEPT AT does not alter this byte. The XB manual however does not document this behaviour at all. If XB were not a dead language that would be a caution signal. It does need to be watched in your programs, if your last CALL KEY was not the key unit you want for ACCEPT AT. On the positive side you can control ACCEPT AT with a prior dummy CALL KEY to ease input for the user. An example is when a program requests input of a filename, setting the key unit to 3 makes letters come out as upper-case while still allowing other characters. Brian Rutherford of HV99 first brought the anomalous behaviour to my attention and has turned up other minor undocumented variations in the use of ACCEPT AT. .

Now that is not too bad, but there is worse to come. Insert a VALIDATE("123") clause in the ACCEPT AT and RUN the program. No problems there with SIZE(3), but SIZE(-3) is trickier. You can not enter invalid characters from the keyboard but unaltered "\_" slip through. The VALIDATE appears to be exercised as characters are entered from the keyboard, and not as the edit buffer contents are transferred into the target variable. The decision to ignore trailing blanks in the input window is taken then, however. Presumably a negative SIZE pre-loads the edit buffer with the screen window contents without doing a VALIDATE check. Ultimately this is not a real problem since the programmer can control what is on the screen before ACCEPT AT is invoked. Once again, the XB manual does not bind ACCEPT AT to work this way.

This behaviour does leave a weak spot in ACCEPT AT which can only be considered as a bug, but not an intractable one. Suppose you have a menu choice of items, say 1-4 by number, with default 1 pre-loaded in the SIZE(-1) window, and a VALIDATE("1234") clause to ensure proper entry for a numeric variable. What can possibly go wrong? An evil-minded program tester would immediately delete the default using FCTN-1. An attempt to enter the blank will then cause the screen to scroll with a WARNING message. This is not a fatal error, but might as well be if your background is a carefully composed graphics screen. The workaround for this problem is not difficult, but the best one also resolves an even worse bug, so I will leave it for a little while. I do consider suppression of error trapping or warning messages by global ON ERROR or ON WARNING to be poor programming practice. The best safety net is one that is never used, only tested.

Now go back to the original sample program and change every A\$ to an array element A\$(2). Default dimensioning will do. Nothing changes. Next alter your A\$(2) in the ACCEPT AT to A\$(1+1). Now it works only if there is also a VALIDATE clause, but the SIZE window is disabled and input can even spill over into the next line. No, it is not useful as a multiline ACCEPT! The solutions to this and the previous problem are the same -- always ACCEPT into a temporary simple string variable, and then process the return, and do not ACCEPT a numeric directly or ACCEPT into an array element with computed index. Both of these problems were turned up by my testing crew during the writing of TEX-BOUNCE, and served as a reminder that program testing should never be left to the author of a program. The same holds true for writers of languages!

Might as well keep on going with the entomology lesson. The sub-program CALL ERR fails to clear errors when the DSR routine cannot find the external device, as in attempts to access an empty disk drive. The work-around this problem is to have a second bash at CALL ERR after further trying for a file on the device which failed to OPEN. The OPEN cannot be CLOSED without crashing the

program or invoking this extra step to flush out the Peripheral Access Block.

The instruction ON BREAK NEXT is useful, particularly in games, for disabling the FCTN-4 (BREAK) key action. However a CALL SOUND with duration greater than 33 over-rides that. Just why is not so far obvious to this outside observer. .

## The BRAIN - a review

by Howie Rosenberg, USA

There has been somewhat of a lack of programs for the TI-99/4A in the category of scientific. The introduction of BRAIN by Datax in no way helps to fill that gap. A demonstration version of this program has been distributed and this review is based on that version. The demonstration was somewhat altered so that Maths routines would give correct answers. The author had introduced a random number into the routines as a feeble method of making the program useless. There was no need to do so as the program is already of very little value. BRAIN is a menu driven collection of Maths and Science routines. The items on the main menu are Annuities, Math and Calculus, Electrodynamics, Trajectories, Conversions, Geometry, Physics, Vectors, Tables, and Help. Selection of many of the items such as Physics, Electrodynamics, and Trajectories yield nothing but a message that this item is unavailable in the demonstration. No matter--there is enough in the demo for the casual user to evaluate. The Maths and Calculus option reveals a rather sparse submenu consisting of Length of Arc, Exponents, Factorials and Logs. Every one of these functions is a single line equation which anyone who really needs to can type in a single line of code with considerably greater speed than the rather slow boot required. I might mention that the slow boot is partially due to a rather lengthy set of call load statements loading assy routines, some paranoid protection (who would really want to pirate this junk) and the piece de resistance the new lower case character set! I have seen a variety of character sets! TI's fix for TI writer followed by the one which many of us use now which originally was part of FAST TERM. Each of these sets is characterized by lower case descenders. We now have the first character set which has lower case ascenders! It is the ugliest character set I have seen. It is downright disturbing.

The menu selection tables yields a submenu containing trig tables, TMS 9900 instruction set, colour codes, metric conversions, ASCII codes, and a table of chemical elements. None of the above really warrants any expenditure by the average user. Those who do need them might find them of value if they were in a rapid, easily accessible format which they are not. Take the element and isotope tables, for example. Upon selection, six screens are presented in succession with a key press to access the next screen. It would make more sense to have access to the required data upon entry of a keyword (a data base!). A program which does no more than provide menu driven solutions to very simple equations and tables is certainly not worth the \$49.95 price tag that DATAX places on this junk. If it were not for their advertisements in Micropendium and the expense they have gone to to distribute this demonstration, I would consider it as a joke. It has become a custom, to include a report card in software reviews. I have been opposed to this practice as grading is subjective unless one can establish a benchmark with which to compare the program. I make an exception in this case because I think that a record may have been established for a commercial product in the TI marketplace. Report Card:

Utility	F
Ease of use	B
Documentation	D (inadequate help files)
Value	F
Final Grade	D-

Conclusion: Of no value at the present price. For say \$10 it may prove to be of minimum use for a very small segment of the user population. .

# Instructions for Up Periscope

© Copyright 1982 by Pewterware.

Object of the game: to sink as much enemy shipping as possible with 12 or 24 torpedoes.

Number of players: This is a one player game.

Uses TI Extended BASIC, if a disk system is attached enter CALL FILES(1) prior to loading the program.

Play: After selecting the level of difficulty and the torpedo load, the screen will display submarine course (000) degrees, depth (100 feet) and speed (4 knots). The sonar scope will display enemy contacts. The true bearing in degrees and range in yards of enemy contacts will be displayed under target and/or escort. Your objective is to make a torpedo approach on the target, sink it with 1 to 3 torpedoes and evade the depth charge attack of the escort.

Your inputs on the keyboard are as follows: (In each case, hold down the appropriate key until the computer acknowledges you).

C - Allows you to change submarine course. Any course from 000 to 359 degrees can be selected. After selection, the course will change 30 degrees at a time until within 30 degrees of the new course. Then it will change 5 degrees at a time.

D - Allows you to change submarine depth. Any depth from 60 to 500 in 10 foot increments can be selected. When coming up to 60 feet, the periscope will come up automatically, and when diving below 60 feet it will lower automatically.

S - Allows you to change submarine speed. Any speed from 0 to 8 knots can be selected. The maximum speed of 8 knots will be reduced when flooding occurs.

P - Allows you to raise or lower the periscope when the submarine is at periscope depth (60 feet). The periscope will automatically track either the target or escort.

T - Sets the periscope to automatically track the target when the periscope is raised.

E - Sets the periscope to track the escort. A screen indicator shows which enemy is being tracked.

1, 2, or 3 - Fires 1, 2 or 3 torpedoes at the enemy being tracked. After firing, the chance of sinking the enemy (from 0% to 100%) will be displayed below the periscope display. The periscope must be raised to fire torpedoes. Naturally your chances of success will be best when firing 3 torpedoes at close range.

Winning: Steer a course to intercept the target ahead of its anticipated track. Be careful not to have the periscope raised when you are at a speed of more than 4 knots and within 4000 yards of the enemy to avoid detection! Prior to firing your torpedoes turn toward the target so that the torpedoes will not have to change course to hit it. After you fire your torpedoes the escort will commence a depth charge attack. Note: It is possible to sink the escort, but extremely difficult; they are fast and sneaky.

Losing: You lose if the escort sinks you. A close depth charge at periscope depth will sink you. A direct hit at a depth of less than 100 feet will sink you. Close depth charges will cause the submarine to start flooding in water. The amount of water will be displayed under "flooding". The deeper the submarine the faster the flooding rate. If flooding exceeds 1000, the submarine will sink, so reduce depth to reduce flooding.

Enemy sonar: During a depth charge attack, the escort will be able to detect you at a depth of 150 feet or less (200 or less for the difficult game; 250 or less for the impossible game). This means that the escort will constantly be changing its course toward you if you are at a shallow depth and are within 4000 yards of the escort during a depth charge attack.

Hint: You do not have enough speed to catch an enemy. You must intercept them ahead of their track to make a successful attack.

```
10 CALL CLEAR :: CALL SCREEN(5)
20 DISPLAY AT(12,9):"UP PERISCOPE" ::
  CALL SOUND(1000,262,5):: CALL SOUND(500,330,5)::
  CALL SOUND(500,392,5)
30 CALL SOUND(1000,440,5):: CALL SOUND(300,330,5)::
  CALL SOUND(2000,440,5,349,10,262,10)::
  CALL SOUND(1,30000,30)
40 CALL CLEAR :: DISPLAY AT(5,2):"SELECT LEVEL OF
  DIFFICULTY" :: DISPLAY AT(10,10):"1-NORMAL"
50 DISPLAY AT(13,10):"2-DIFFICULT" ::
  DISPLAY AT(16,10):"3-IMPOSSIBLE"
60 DISPLAY AT(20,10):"1" ::
  ACCEPT AT(20,10)SIZE(-1)VALIDATE(DIGIT)BEEP:DF :: IF
  DF>3 OR DF<1 THEN 60
70 DISPLAY AT(5,5)ERASE ALL:"SELECT TORPEDO LOAD" ::
  DISPLAY AT(8,10):"12 OR 24"
80 ACCEPT AT(11,13)VALIDATE(DIGIT)BEEP:TORP ::
  IF TORP<>12 AND TORP<>24 THEN 80 :: TP=TORP
90 CALL CLEAR :: CALL SCREEN(2):: FOR I=9 TO 14 ::
  CALL COLOR(I,2,2):: NEXT I
95 RESTORE 20400 :: FOR I=1 TO 3 :: READ E$(I),T$(I)::
  NEXT I
100 TONS=0 :: H=0 :: MS=8 :: E=0 :: WATER=0 :: WR=0 ::
  CALL MAGNIFY(4)
105 DEF V=2+(S$="U"):: DEF TD=10*INT(25*SQR(R1)) ::
  DEF ED=10*INT(25*SQR(R2))
110 R=81 :: C=169 :: S$="D" :: RANDOMIZE ::
  CALL COLOR(0,2,6):: FOR I=2 TO 8 ::
  CALL COLOR(I,2,6):: NEXT I
150 DEF M=INT(MINUTE/10)::
  DEF TR=2*(SPD*COS(CRS*PI/180)-TSPD*COS(TCRS*PI/180))
180 ON WARNING NEXT ::
  DEF TBRG=INT(180*ATN((TCP-C)/(R-TRP))/PI)-
  180*(R<TRP)+360*(R>=TRP)*(C>TCP)
190 SPD=4 :: NS=4 :: CRS=0 :: NC=0 :: DEP=100 :: ND=100
  :: DEF
  TC=2*(TSPD*SIN(TCRS*PI/180)-SPD*SIN(CRS*PI/180))
195 DEF ER=2*(-ESPD*COS(ECRS*PI/180)+SPD*
  COS(CRS*PI/180)):: DEF EC=2*(-SPD*SIN(CRS*PI/180)+
  ESPD*SIN(ECRS*PI/180))
200 DEF EBRG=INT(180*ATN((ECP-C)/(R-ERP))/PI)-
  180*(R<ERP)+360*(R>=ERP)*(C>ECP)
500 RESTORE 540 :: FOR I=96 TO 114 :: READ A$,B$ ::
  CALL CHAR(I,A$):: CALL CHAR(I+24,B$):: NEXT I
510 FOR I=96 TO 113 :: READ J,K,A,B ::
  CALL HCHAR(J,K,I):: CALL HCHAR(A,B,I+24):: NEXT I
530 FOR I=5 TO 12 :: READ J,K :: CALL HCHAR(I,J,114,K)
  :: CALL HCHAR(25-I,J,138,K):: NEXT I
540 DATA COC080808080808,0101010101010303,
  FOEOEOEOEOCOCOC,030303070707070F,FFFEFCFCF8F8F0F,
  OFOF1F1F3F3F7FFF
560 DATA F8FOEOEOCOCOC,0001030307070F1F,
  FFFFFFFF7FFF7FFF,FEFCFOC08,
  00000001030F3F7F
570 DATA FFFFFFFF7FFF7FFF,01030F3FFFFFFF,FFFFFFF8,
  0000000001FFFFFF,FFC,00000000000003FF,FF08,
  000000000000COFF
590 DATA FFFF1F01,0000000080F8FFFF,FFFFFFF3F0F0301,
  80C0F0FCFFFFFFF7F3F0F0301,00000080C0F0FCFE
600 DATA FFFFFFFF7F3F,FCFEFFFFFFF7F3F,
  1F0F0707030301,0080C0C0EOEOFOF8,FF7F3F3F1F1F0F0F,
  F0F0F8F8FCFCFEFF
620 DATA OF07070707030303,COCOC0EOEOEOEO,
  0303010101010101,808080808080COC,0,0
630 DATA 10,16,15,31,9,16,16,31,8,16,17,31,7,17,18,30,
  6,17
640 DATA 19,30,6,18,19,29,5,19,20,28,5,20,20,27,5,21,
  20,26
650 DATA 5,26,20,21,5,27,20,20,5,28,20,19,6,29,19,18,6,
  30
660 DATA 19,17,7,30,18,17,8,31,17,16,9,31,16,16,10,31,
  15,16
670 DATA 22,4,19,10,18,12,17,14,17,14,17,14,16,16,16,16
```



```

4510 IF SUNK=1 THEN 4545 ::
  CALL CHAR(140,T$(3+(TD>12000)+(TD>4000))): IF T=2
  THEN 4525
4520 CALL SPRITE(#1,140,2,65,169):: GOTO 4550
4525 IF ABS(TBRG-BRG)>10 AND ABS(TBRG-BRG)<350 THEN 4545
4530 IF ABS(TBRG-BRG)<=10 THEN BRG1=TBRG-BRG ELSE
  BRG1=ABS(TBRG-BRG)-360
4540 CALL SPRITE(#1,140,2,65,169+8*BRG1):: GOTO 4550
4545 CALL DELSPRITE(#1)
4550 IF SUNK=2 THEN 4610
4560 CALL CHAR(36,E$(3+(ED>12000)+(ED>4000))): IF
  T=2 THEN 4590
4565 IF ABS(EBRG-BRG)>10 AND ABS(EBRG-BRG)<350 THEN 4600
4570 IF ABS(EBRG-BRG)<=10 THEN BRG1=EBRG-BRG ELSE
  BRG1=ABS(EBRG-BRG)-360
4580 CALL SPRITE(#2,36,2,65,169+8*BRG1):: GOTO 4610
4590 CALL SPRITE(#2,36,2,65,169):: GOTO 4610
4600 CALL DELSPRITE(#2)
4610 DISPLAY AT(3,23):SEG$(STR$(1000+BRG),2,3)::
  IF (ED<4000 OR TD<4000)AND SPD>4 THEN 12120
4620 RETURN
8000 TSPD=7+2*DF+INT(5*RND):: ESPD=TSPD :: DC=0 ::
  GOSUB 14050
8010 IBRG=(360*RND)*PI/180 :: TRP=R+60*COS(IBRG)::
  TCP=C+60*SIN(IBRG):: CALL SPRITE(#7,116,V,TRP,TCP)
8020 TCRS=TBRG+140+INT(80*RND):: ECRS=TCRS
8030 CALL SPRITE(#8,116,V,TRP-10*COS(TCRS*PI/180),
  TCP+10*SIN(TCRS*PI/180)):: SUNK=0 :: RETURN
8500 TCRS-TBRG :: ECRS=TCRS :: TSPD=11+2*DF :: ESPD=TSPD
:: RETURN
9000 E=E+1 :: CALL DELSPRITE(E):: IF E<8 THEN 9000
9005 CALL CLEAR :: CALL SCREEN(6)::
  DISPLAY AT(5,2):"TOTAL TONNAGE SUNK";TONS
9010 DISPLAY AT(10,7):"YOUR RATING IS:" ::
  IF TONS<=2500*TP THEN 9040 ELSE IF TONS<=5000*TP
  THEN 9030
9020 DISPLAY AT(13,4):"LEVEL";DF;"SUBMARINE ACE" ::
  GOTO 9050
9030 DISPLAY AT(13,8):"LEVEL";DF;"EXPERT" :: GOTO 9050
9040 DISPLAY AT(13,7):"LEVEL";DF;"RECRUIT"
9050 DISPLAY AT(24,1):"PRESS ENTER FOR ANOTHER GAME"
9060 CALL KEY(0,K,S):: IF S=0 THEN 9060 :: IF K=13 THEN
  40 :: END
10000 CALL MOTION(#7,TR,TC):: CALL MOTION(#7,0,0)::
  CALL MOTION(#8,ER,EC):: CALL MOTION(#8,0,0)
10010 CALL DISTANCE(#6,#7,R1):: CALL DISTANCE(#6,#8,R2)
:: CALL POSITION(#7,TRP,TCP)::
  CALL POSITION(#8,ERP,ECP)
10015 IF TD<16000 THEN 10018 :: CALL DELSPRITE(#7)::
  DISPLAY AT(20,3)SIZE(9):"":: GOSUB 14000
10018 IF ED<16000 THEN 10030 :: CALL DELSPRITE(#8)::
  DISPLAY AT(24,3)SIZE(9):"":: GOSUB 14050
10020 IF SUNK=3 OR (ED>=16000 AND TD>=16000)OR(SUNK=1 AND
  ED>=16000)OR(SUNK=2 AND TD>=16000)THEN 8000
10030 IF DC>0 AND DEP=60 AND ED<101+150*DF THEN 10500
10040 IF DC>0 AND DEP<100+50*DF AND ED<8000 THEN
  ECRS=EBRG-180-360*(EBRG<180)
10050 IF SUNK=1 OR TD>=16000 THEN 10070 ::
  DISPLAY AT(20,3)SIZE(9)BEEP:
  SEG$(STR$(TBRG+1000),2,3)&" "&STR$(TD)
10070 IF SUNK=2 OR ED>=16000 THEN 10130 ::
  DISPLAY AT(24,3)SIZE(9)BEEP:
  SEG$(STR$(EBRG+1000),2,3)&" "&STR$(ED)
10130 IF SUNK=0 AND SPD>8 AND(ED<4000 OR TD<4000)AND
  DEP=60 THEN GOSUB 12120
10140 IF S$="U" THEN 4500 :: RETURN
10500 CALL LOCATE(#8,R,C):: R2=0 :: GOTO 10050
12000 IF T=2 THEN 12030
12010 CHANCE=100-SQR(TD)-ABS(TBRG-CRS):: GOTO 12040
12030 CHANCE=50-SQR(ED)-ABS(EBRG-CRS)
12040 HIT(1)=100*RND :: CALL SOUND(500,-5,10)::
  CALL SPRITE(#3,116,15,140,158,-6,1):: TORP=TORP-1
12045 IF K>49 AND TORP>0 THEN 12050
12046 IF CHANCE>0 THEN 12047 :: CHANCE=0
12047 DISPLAY AT(22,21):INT(CHANCE):: FOR I=1 TO 70 ::
  NEXT I :: GOTO 12070
12050 HIT(2)=100*RND :: CALL SOUND(500,-5,10)::
  CALL SPRITE(#4,116,15,140,169,-6,0):: TORP=TORP-1
12055 IF K>50 AND TORP>0 THEN 12060
12056 IF CHANCE>0 THEN 12057 :: CHANCE=0
12057 DISPLAY AT(22,21):INT(100-(100-CHANCE)^2)/100)::
  FOR I=1 TO 30 :: NEXT I :: GOTO 12070
12060 HIT(3)=100*RND :: CALL SOUND(500,-5,10)::
  CALL SPRITE(#5,116,15,140,180,-6,-1):: TORP=TORP-1
12063 IF CHANCE>0 THEN 12065 :: CHANCE=0 continued on page 18

```

# TI-Base Tutorial #6

by Martin Smoley, NorthCoast 99ers  
© Copyright 1988 by Martin A. Smoley

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as it is free.

\*\*\* I am Sorry! \*\*\*

Last month I left out the program segment listed below. I was looking over the tutorial, well after the newsletter had gone to the printer, and near the top of FNDPRNT1 I saw it. It sticks out like a sore thumb, DO DSK2.INFSCR2. I instantly knew that it was not in the tutorial. "What a dumb thing to do." So here it is and I hope this did not wreck your holiday computing.

```
CLEAR
WRITE 3,9,"This section locates a  "
WRITE 5,9,"record using the NM field."
WRITE 7,9,"It then displays the  "
WRITE 9,9,"name and address and asks"
WRITE 11,9,"how many labels you want."
WRITE 13,9,"It will find as many"
WRITE 15,9,"records as you wish."
WRITE 17,9,"** NEWNAMES has 5 RECORDS."
RETURN
*
* INFSCR2      Save as INFSCR2/C
* *****    Info Screen 2 12/1/88
*
*****
```

## TI-Base Version 2.0

As I stated in December I am switching to TI-Base Version 2.0. Ver. 2.0 still has a couple of small bugs in it, but it is almost bug free and I expect the CHANGE commands to arrive at any minute. Also, I would add that it already works better, runs and loads faster, does more than Ver. 1.02 and the Manual has twice as much information (plus it is easier to read). The upgrade from Ver. 1.02 is only \$7.95 plus your original system disks, so you should get yours as soon as possible. I do suggest that you keep a copy of the old version. It will probably come in handy at some time.

Let's get started. Two items that caught my eye immediately were READSTRING and the use-ability of .DATE. The READ command is still in use. It will accept the input of numbers with no qualms. READ will also accept the input of characters if you place your data in quotes. So to answer the question, CONTINUE? Y/N, you would have to answer "Y" or "N", including the quotes. If you use the new READSTRING, your answer would be Y or N without the quotes. This makes things a lot easier.

```
* Copyright 1989 By Martin A. Smoley *
LOCAL LYDT C 2
LOCAL TYDT C 2
REPLACE LYDT WITH "88"
REPLACE TYDT WITH "89"
DOCASE
CASE MM="01"
REPLACE CUTOFF WITH LYDT | "/09"
REPLACE PRDT WITH TYDT | "/01"
BREAK
CASE MM="02"
REPLACE CUTOFF WITH LYDT | "/10"
REPLACE PRDT WITH TYDT | "/02"
BREAK
CASE MM="03"
REPLACE CUTOFF WITH LYDT | "/11"
REPLACE PRDT WITH TYDT | "/03"
BREAK
```

```
CASE MM="04"
REPLACE CUTOFF WITH LYDT | "/12"
REPLACE PRDT WITH TYDT | "/04"
BREAK
CASE MM="05"
REPLACE CUTOFF WITH LYDT | "/01"
REPLACE PRDT WITH TYDT | "/05"
BREAK
CASE MM="06"
REPLACE CUTOFF WITH LYDT | "/02"
REPLACE PRDT WITH TYDT | "/06"
BREAK
CASE MM="07"
REPLACE CUTOFF WITH LYDT | "/03"
REPLACE PRDT WITH TYDT | "/07"
BREAK
CASE MM="08"
REPLACE CUTOFF WITH LYDT | "/04"
REPLACE PRDT WITH TYDT | "/08"
BREAK
CASE MM="09"
REPLACE CUTOFF WITH LYDT | "/05"
REPLACE PRDT WITH TYDT | "/09"
BREAK
CASE MM="10"
REPLACE CUTOFF WITH LYDT | "/06"
REPLACE PRDT WITH TYDT | "/10"
BREAK
CASE MM="11"
REPLACE CUTOFF WITH LYDT | "/07"
REPLACE PRDT WITH TYDT | "/11"
BREAK
CASE MM="12"
REPLACE CUTOFF WITH LYDT | "/08"
REPLACE PRDT WITH TYDT | "/12"
BREAK
ENDCASE
RETURN
*
* RSTRCS1      Save as RSTRCS1/C
* *****    DOCASE for PRSTR1 01/02/89
```

The CF for this month is another club roster printout program. I say another because there was a club roster segment included in last month's system CF. This one can be used alone by typing DO DSK2.PRSTR2 <E>, at the DP or you can substitute it for the roster segment in the system. In any case PRSTR2 needs to have PREP1 and FIN1 available to it on disk 2 when it runs. These two segments were included in last month's system program. In this tutorial I present PRSTR2, RSTRCS1 and RSTR2WH. The complete set works like this. You have a DB like NEWNAMES that contains your club membership list. The list is sorted by LN (Last Name) and each member's renewal or expiration date is kept as YY/MM (Year/Month) in the data field named XP. In order for this CF to work properly you must enter the date properly when you start up TIB. When TIB first boots up and asks you to enter the date it is in the form of MM/DD/YY, or Month/Day/Year. January 5, 1989 is the first month, the fifth day, of 1989, or 01/05/89. It must be complete (01/05/89). Do not leave out the zeros. If the date entry is correct and you type DO DSK2.PRSTR2, this is how the CF will work for you. It creates the variable MM with only 2 spaces available. It sticks .DATE. into MM. Then it throws away all but the first two characters, the Month part of the date. It takes MM and DOES RSTRCS1. This CF executes only the 2 lines

```
DO DSK2.PREP1
SET PAGE=000
CLEAR
LOCAL COUNT N 4 0
LOCAL TEMP C 79
LOCAL BLNK C 1
LOCAL CUTOFF C 5
LOCAL PRDT C 5
LOCAL MM C 2
REPLACE MM WITH .DATE.
USE NEWNAMES
DO DSK2.RSTRCS1
WRITE 10,4,"Set Printer + press ENTER"
READSTRING 10,30,TEMP
CLEAR
WRITE 10,12,"Printing Roster"
```

continued on page 18

# Subject Index for 1989

Title	Description	Author	Page No.	Title	Description	Author	Page No.
32K memory expansion	Hardware project	Amadio,Lou	12.07	From the trenches	General interest	Kanitz,Werner	10.30
	Hardware project	Mudie,Ross	12.09	Full day workshop	Club news	Sheehan,Craig	6.04
					Club news	Sheehan,Craig	11.04
Advanced Diagnostics review	Software review		8.29		Software hints	Swedlow,Jim	2.13
Adventure games	General interest	Saunders,Larry	6.19	Funnelweb configuration			
Agenda for AGM, 1989	Club news	Phillips,Terry	1.01				
Alternative micro show	General interest	Shaw,Stephen	1.15	Game of Wit	Software review	Lang,Chris	10.16
Anatomy of a pirate	General interest	Paine,John	7.30	Games information	General interest	Brown,Robert	2.19
Answer is abc, question?	General interest	Takach,Ben	10.03		General interest	Judd,Stephen	2.19
Archiver III version 3.02	Software review	Boone,Barry	2.27		Witness, Karate chall.	Brown,Robert	12.13
Assembler tutorial weekend	General interest	Mudie,Ross	5.07		Zork 1	Brown,Robert	5.17
	General interest	Mudie,Ross	6.05		Zork 1	Judd,Stephen	5.17
	General interest	Mudie,Ross	8.02		Zork 2	Brown,Robert	6.15
AT music card	Hardware project	Schubert,Peter	2.28		Zork 2	Judd,Stephen	6.15
Author index volume 7, 1988	General interest		1.08		Zork 3	Brown,Robert	9.17
					General interest	Schreiber,Rolf	5.04
Backing up the hard disk	General interest	Takach,Ben	9.22	Getting your money's worth?	General interest	Murrell,William	11.06
Beginner's corner	General interest	Bobbitt,Chris	9.08	Glossarial on acronyms	General interest		
Beyond video chess	Software review	Saunders,Larry	6.12	Grade standardizer using X/B	Software hints	Relyea,Robert	5.27
Book reviews	General interest	Shaw,Stephen	6.12	Graphics interchange format	Software hints		7.31
Buckaroo Banzai	Adventure hints	Saunders,Larry	5.13				
	Adventure hints	Saunders,Larry	6.24	Handy hints	Hardware, software hint	Mudie,Ross	5.32
Bug report	DM1000 version 3.5	Trott,Geoff	4.06	Hard disk drive - guided tour	Software hints	Christensen,Garry	9.27
	Myarc hard controller	Trott,Geoff	9.04	Hard disk drive data	Hardware review	Paine,John	9.23
	TI-Writer and formatter	Trott,Geoff	5.29	HardMaster	Software review	Christensen,Colin	9.25
c99 printer setup	Software hints	Trott,Geoff	6.33	I/O interface PCB hints	Hardware project	Amadio,Lou	11.08
c99 quick reference	Software hints		12.24	Infidel by Infocom	Adventure solution	Ian	2.07
c99 tutorial	Software hints	Sheehan,Craig	8.14	Interface IBM style keyboard	Hardware project	Wilkinson,Derek	9.05
	Software hints	Sheehan,Craig	9.12	Interfacing to RS232 PIO port	Hardware project	Trott,Geoff	4.09
	Software hints	Sheehan,Craig	10.12				
	Software hints	Sheehan,Craig	12.23	Joystick interface	Hardware project	Amadio,Lou	10.05
Calendar programs	General interest	Robinson,Adrian	12.21	Joytalk, low cost printer port	Hardware review	Mudie,Ross	3.24
Cassette finder review	Software review	Revitt,Steve	6.08		Software hints	Mudie,Ross	4.10
Clock programs with hard disk	Software hints	Takach,Ben	8.15				
Co-ordinators report	General news	Warburton,Dick	3.02	Kaleidawriter data-bank	Hardware news	Cartwright,Colin	4.29
	General news	Warburton,Dick	4.02				
	General news	Warburton,Dick	5.02	Legends	Adventure hints	Saunders,Larry	4.13
	General news	Warburton,Dick	6.02		Adventure hints	Saunders,Larry	8.21
	General news	Warburton,Dick	7.02		Adventure hints	Saunders,Larry	8.21
	General news	Warburton,Dick	8.02	Lego, not Logo turtle	Hardware project	Mudie,Ross	6.31
	General news	Warburton,Dick	9.02	Letter to editor	Fairware payments	Amadio,Lou	5.28
	General news	Warburton,Dick	10.02		Leaving group	Bull,Greg	12.02
	General news	Warburton,Dick	11.02		Life membership	Mudie,Ross	3.03
	General news	Warburton,Dick	12.02		Magic squares	Peterson,Jim	6.03
Communicators	BBS information	Mudie,Ross	2.05		Merry Christmas	Anderson,Shane	2.16
	BBS information	Mudie,Ross	3.05		Modems	Mudie,Ross	6.05
	BBS information	Mudie,Ross	7.11		New directors	Ryan,John	2.03
	BBS information	Mudie,Ross	11.05		Participation	Anonymous	4.04
	BBS information	Mudie,Ross	11.05		Regional groups	Anonymous	4.04
Communicators extra	Software review	Saunders,Larry	3.19		Symposium for TI99/4A	Amadio,Lou	7.02
Configuring PrEditor	Hardware project	Mudie,Ross	2.31		Views of TISHUG	Ryan,John	4.04
Controlling a train set	Software hints	Harris,D.N	9.33	Magoo's mumblings	Software hints	Magoo,Mr	5.21
Conversion tables	Software hints	Amadio,Lou	4.23	Making your own dictionaries	Software hints	Trott,Geoff	8.23
Create characters for printer					Software hints	Trott,Geoff	9.30
					Software hints	Trott,Geoff	10.31
Deadline by Infocom	Adventure solution	Ian	2.07	Mass transfer	Software review	Olson,Stuart	7.12
Death of a computer	General interest	Paine,John	3.07	MCOPY or disk drive saver	Software review	Arnold,Tom	5.07
Direct I/O for PEB cards	Hardware project	Amadio,Lou	7.05	Mechatronic 80 column card	Hardware review	Takach,Ben	8.31
	Hardware project	Amadio,Lou	10.05	MiniMemory meandering memories	Software hints	Shaw,Stephen	7.08
	Hardware project	Amadio,Lou	11.07	Minutes of 1988 AGM	Club news	Phillips,Terry	1.02
Direct I/O interface, errata	Hardware project	Amadio,Lou	8.05	Modems	General interest	Harris,D.N	5.33
Disassembling Extended BASIC	Software hints	Mudie,Ross	4.22	Myarc hard disk controller	Hardware review	Amadio,Lou	9.21
Disassembling X/B correction	Software correction	Mudie,Ross	5.10	Myarc rumours	General interest	Schreiber,Rolf	5.26
Disk Hacker documentation	Software review	McGovern,Will	2.10	Myarc, Geneve update	Hardware news	Hope,Greg	3.03
Disk controls	Software hints	Ballmann,Michael	10.27				
Disk utilities, X/B tips	Software review, hints	Swedlow,Jim	11.23	New TI99/4A	General interest	Saunders,Larry	8.07
					Software review	Saunders,Larry	12.04
Electronic Bible	General interest		6.05	New programs and games	General interest	Bobbitt,Chris	9.32
Emulate file pointer	Software hints	Christensen,Garry	9.22	News from around the world	General interest	Amadio,Lou	3.29
Expanding X/B's powers	Software hints	Caron,David	4.08	Newsletter update	General interest	Amadio,Lou	4.30
	Software hints	Caron,David	6.11		General interest	Amadio,Lou	5.34
	Software hints	Caron,David	8.04		General interest	Amadio,Lou	6.34
	Software hints	Caron,David	10.04		General interest	Amadio,Lou	8.10
Expanding your TI99/4A	Hardware review	Saunders,Larry	4.11		General interest	Amadio,Lou	9.34
Extended BASIC colours	Software hints	Amadio,Lou	2.09		General interest	Amadio,Lou	11.34
Extended BASIC screen colours	Software hints		12.12		General interest	Amadio,Lou	12.30
Extended BASIC subprograms	Software hints	Shaw,Stephen	10.15				
Extended BASIC tips	Software hints	Swedlow,Jim	6.10	P-GRAM card and Geneve	Software review	Saunders,Larry	5.25
Extended BASIC tutorial	Software hints	McGovern,Tony	9.09	PE box unregulated voltages	Hardware project	Takach,Ben	3.22
	Software hints	McGovern,Tony	10.07	Plotic	General interest	Harris,D.N	12.20
	Software hints	McGovern,Tony	11.13	Power supply for hard disk	Hardware project	Amadio,Lou	9.06
	Software hints	McGovern,Tony	12.05	PrEditor	Software review	Saunders,Larry	3.17
Extended display package	Software review	Sheehan,Craig	2.07	Press	Software review	Saunders,Larry	5.08
	Software review	Sheehan,Craig	3.06	Press, the next generation	Word processing	Bobbitt,Chris	2.39
	Software review	Sheehan,Craig	4.07	Printer multifont control	Software hints	Amadio,Lou	5.26
	Software review	Sheehan,Craig	5.09	Printer problems	Software hints	Peterson,Jim	11.31
				Printing 132 columns with ATMF	Software hints	Trott,Geoff	5.12
Fairware author of month	Ottawa user group	Trott,Geoff	8.03	Problem solving procedures	General interest		8.24
	Ottawa user group	Trott,Geoff	9.02	Program to type in	Animal learner		10.18
	Stuart Olsen	Trott,Geoff	10.02		Artist board		2.23
	Stuart Olsen	Trott,Geoff	11.03		Brain teaser		2.20
	Tony McGovern	Trott,Geoff	7.20		County fair derby		2.21
Fast terminal program	Software review	Charlton,Paul	6.29		Grade calculator	Demming,Bob	1.15
Floppy disk power supply	Hardware project	Amadio,Lou	8.05		Kingdom of Sumaria		10.17
Forth dimension	Software hints	Stanford,Jeff	12.27		Magic squares		3.14
Forth tidbits 1 to 8	Software hints	Winkler,Lutz	10.23		Miniature golf		3.15
Forth tidbits 7, 8 and 9	Software hints		11.27		Regression analysis		4.14
Forth to you too!	Software hints		9.33		Statistical parameters	Livingstone,E	11.17
	Software hints		10.23		Trigonometry	Jingozian,Philip	4.15
	Software hints		11.27		Wankapillar	Sumner,Mark	12.15
	Software hints		12.27	Programming hint: not BASIC	Software hints	Shaw,Stephen	6.20
From the bulletin board	Mail to all		2.05	Programs from TISHUG library	Software review	Shaw,Stephen	3.30
	Mail to all		3.05		Software review	Shaw,Stephen	6.34
	Mail to all		4.06		Software review	Shaw,Stephen	7.11
	Mail to all		5.10	Project Management	General interest		12.14
	Mail to all		6.05	Public domain releases	Software review	Saunders,Larry	4.02
	Mail to all		7.11	Club library	Club library	Welham,Warren	3.27
	Mail to all		9.31	Publications library report	Club library	Welham,Warren	5.08
	Mail to all		10.33				
	Mail to all		11.05	Questprobe with spiderman	Adventure hints	Saunders,Larry	3.13
	Mail to all		12.12	Quick hard disk back-up	Software hints	Christensen,Garry	9.21

Title	Description	Author	Page No.	Title	Description	Author	Page No.
Rambles from UK	General interest	Shaw, Stephen	2.25	TI-Base tutorial	Data base	Smoley, Martin	8.11
	General interest	Shaw, Stephen	6.32		Data base	Smoley, Martin	9.13
	General interest	Shaw, Stephen	11.12		Data base	Smoley, Martin	10.09
	General interest	Shaw, Stephen	12.29		Data base	Smoley, Martin	11.21
RAMdisk or hard disk?	Hardware review	Amadio, Lou	10.29		Data base	Smoley, Martin	12.25
RAMdisk power supply problems	Hardware project	Trott, Geoff	9.06	TI-Writer bug	Word processing	Peterson, Jim	3.26
RAMdisk, another view	General interest	Schreiber, Rolf	6.10	TI-Writer, Funnelweb tutorial	Word processing	Hansen, Tor	6.06
Regional group reports	General interest		2.43		Word processing	Verhey, Dwayne	6.06
	General interest		3.31	Tidbits 8	Software hints	Bowmer, Wade	7.21
	General interest		4.31	Tiger does not sleep tonight	General interest	Scheidemantle, Paul	6.13
	General interest		5.35	Tiger's tale	General interest	Brashear, Harry	4.21
	General interest		6.35		General interest	Brashear, Harry	10.20
	General interest		7.35	Tigercub public domain catalog	Software review	Peterson, Jim	5.18
	General interest		8.35		Software review	Peterson, Jim	5.19
	General interest		9.35		Software review	Peterson, Jim	8.17
	General interest		10.35		Software review	Peterson, Jim	11.25
	General interest		11.35	Tigercub software catalog	Software review	Peterson, Jim	4.27
	General interest		12.31		Software review	Peterson, Jim	6.17
Reliable LOAD interrupt	Hardware project	Amadio, Lou	3.21		Software review	Peterson, Jim	8.19
	Hardware project	Trott, Geoff	3.21		Software review	Peterson, Jim	11.15
				Tigercub tips index	Software index	Peterson, Jim	6.16
Secretary's notebook	Club news	Phillips, Terry	2.02	TIMES software competition	General interest	Gleed, Peter	5.02
	Club news	Phillips, Terry	3.03	Tips from the tigercub #29	Software hints	Peterson, Jim	4.25
	Club news	Phillips, Terry	4.03	Tips from the tigercub #30	Software hints	Peterson, Jim	5.15
	Club news	Phillips, Terry	5.03	Tips from the tigercub #31	Software hints	Peterson, Jim	6.25
	Club news	Phillips, Terry	6.03	Tips from the tigercub #32	Software hints	Peterson, Jim	7.15
	Club news	Phillips, Terry	7.03	Tips from the tigercub #33	Software hints	Peterson, Jim	8.25
	Club news	Phillips, Terry	8.03	Tips from the tigercub #34	Software hints	Peterson, Jim	9.19
	Club news	Phillips, Terry	9.03	Tips from the tigercub #35	Software hints	Peterson, Jim	10.13
	Club news	Phillips, Terry	10.03	Tips from the tigercub #36	Software hints	Peterson, Jim	11.19
	Club news	Phillips, Terry	11.03	Tips from the tigercub #37	Software hints	Peterson, Jim	12.19
	Club news	Phillips, Terry	12.03	Tips from the tigercub #53	Software hints	Peterson, Jim	2.29
Sector sharing	Software hints	Shafer, Mark	6.09	Tips from the tigercub #54	Software hints	Peterson, Jim	3.25
Sending mail on TEXPAC BBS	Software review	Mudie, Ross	4.05	Tips from the tigercub #55	Software hints	Peterson, Jim	5.23
Setup for smart modem	General interest	Saunders, Larry	10.33	Tips from the tigercub #56	Software hints	Peterson, Jim	6.27
Shahzada endurance test	General interest	Mudie, Ross	11.09	Tips from the tigercub #57	Software hints	Peterson, Jim	7.19
Special character key emulation	Hardware project	Amadio, Lou	8.28	Tips from the tigercub #58	Software hints	Peterson, Jim	10.21
Standard marks with Multiplan	Software hints	Relyea, Robert	5.27	TiSHUG in 1989	General interest	Takach, Ben	3.02
Starting a hard disk	Hardware review	Trott, Geoff	4.19	TiSHUG shop	For sale	Carr, Steven	2.03
Starting using TI99/4A	Software hints	Shaw, Stephen	12.17		For sale	Carr, Steven	5.22
Strings in Extended BASIC	Software hints	Sheehan, Craig	8.27		Club news	Trott, Geoff	5.06
				TiSHUG software awards	Club software	Phillips, Terry	2.04
Techo time	Hardware project	Amadio, Lou	3.21	TiSHUG software column	Club software	Phillips, Terry	3.04
	Hardware project	Amadio, Lou	4.09		Club software	Phillips, Terry	4.03
	Hardware project	Amadio, Lou	5.04		Club software	Phillips, Terry	5.05
	Hardware project	Amadio, Lou	6.07		Club software	Phillips, Terry	6.03
	Hardware project	Amadio, Lou	7.05		Club software	Phillips, Terry	7.04
	Hardware project	Amadio, Lou	8.05		Club software	Phillips, Terry	9.04
	Hardware project	Amadio, Lou	9.05		Club software	Phillips, Terry	10.03
	Hardware project	Amadio, Lou	10.05		Club software	Phillips, Terry	11.03
	Hardware project	Amadio, Lou	11.07		Club software	Phillips, Terry	12.03
	Hardware project	Amadio, Lou	12.07	Title index volume 7, 1988	General interest		1.03
Ten commandments for technician	General interest		11.08	Transfers, cassette and disk	Software hints	Rebel, Eric-Paul	9.32
TEXPAC BBS version 4F	BBS Information	Mudie, Ross	12.04	Trials with Myarc DM5	General interest	Takach, Ben	10.26
They're off	General interest	Trott, Geoff	2.01	Trojan horse	General interest	Dodd, Mike	3.27
	General interest	Trott, Geoff	3.01	Turbo pasc 99 tutorial	Software review	Shaw, Stephen	5.31
	General interest	Trott, Geoff	4.01	Typewriter 99	Software review	Saunders, Larry	5.11
	General interest	Trott, Geoff	5.01				
	General interest	Trott, Geoff	6.01	What is GIF?	General interest	Telford, Darren	5.30
	General interest	Trott, Geoff	7.01	Wire I/O update	Hardware project	Mudie, Ross	3.24
	General interest	Trott, Geoff	8.01	Working with numbers	Software hints	Nollan, Joe	12.28
	General interest	Trott, Geoff	9.01				
	General interest	Trott, Geoff	10.01	XXB documentation	Software review	Traver, Barry	11.30
	General interest	Trott, Geoff	11.01				
	General interest	Trott, Geoff	12.01	Younger set	Program, adventure	Maker, Vincent	4.06
TI bits 6, 7, 10, 11	Software hints	Swedlow, Jim	2.15		Program, adventure	Maker, Vincent	11.33
TI Planner, second thoughts	Software hints	Shaw, Stephen	2.24		Program, adventure	Maker, Vincent	12.18
TI Runner level editor	Software review	Shaw, Stephen	7.17		Programs	Maker, Vincent	2.09
TI-Artist upgrade	General interest	Saunders, Larry	5.08		Programs	Maker, Vincent	3.28
TI-Base tutorial	Data base	Phillips, Terry	5.24		Programs	Maker, Vincent	7.09
					Programs	Maker, Vincent	8.22
					Programs	Maker, Vincent	10.33

# Author Index for 1989

Author/Source	Title	Description	Page No.	Author/Source	Title	Description	Page No.
Amadio, Lou	32K memory expansion	Hardware project	12.07	Arnold, Tom	MCOPY or disk drive saver	Software review	5.07
	Create characters for printer	Hardware project	4.23	Hamilton, Canada			
	Direct I/O for PEB cards	Hardware project	7.05	Ballmann, Michael	Disk controls	Software hints	10.27
	Direct I/O for PEB cards	Hardware project	10.05	USA	Beginner's corner	General interest	9.08
	Direct I/O for PEB cards	Hardware project	11.07	Bobbitt, Chris	News from around the world	General interest	9.32
	Direct I/O interface, errata	Hardware project	8.05	Asgard News, USA	Press, the next generation	Word processing	2.39
	Extended BASIC colours	Software hints	2.09				
	Floppy disk power supply	Hardware project	8.05	Boone, Barry	Archiver III version 3.02	Software review	2.27
	I/O interface PCB hints	Hardware project	11.08	OK, USA	Tidbits 8	Software hints	7.21
	Joystick interface	Hardware project	10.05	Bowmer, Wade	Tiger's tale	General interest	4.21
	Letter to editor	Firmware payments	5.28	Brashear, Harry	Tiger's tale	General interest	10.20
	Letter to editor	Symposium for TI99/4A	7.02	Asgard News, USA	Tiger's tale	General interest	2.19
	Myarc hard disk controller	Hardware review	9.21	Brown, Robert	Games information	General interest	12.13
	Newsletter update	General interest	3.29		Games information	Witness, Karate chall.	5.17
	Newsletter update	General interest	4.30		Games information	Zork 1	5.15
	Newsletter update	General interest	5.34		Games information	Zork 2	6.15
	Newsletter update	General interest	6.34		Games information	Zork 3	9.17
	Newsletter update	General interest	8.10		Games information	Leaving group	12.02
	Newsletter update	General interest	9.34				
	Newsletter update	General interest	11.34				
	Newsletter update	General interest	12.30	Caron, David	Expanding X/B's powers	Software hints	4.08
	Power supply for hard disk	Hardware project	9.08	USA	Expanding X/B's powers	Software hints	6.11
	Printer multifont control	Software hints	5.26		Expanding X/B's powers	Software hints	8.04
	RAMdisk or hard disk?	Hardware review	10.29		Expanding X/B's powers	Software hints	10.04
	Reliable LOAD interrupt	Hardware project	3.21	Carr, Steven	TiSHUG shop	For sale	2.03
	Special character key emulation	Hardware project	8.28		TiSHUG shop	For sale	5.22
	Techo time	Hardware project	3.21	Cartwright, Colin	Kateidawriter data-bank	Hardware news	4.29
	Techo time	Hardware project	4.09	Adelaide, SAUST			
	Techo time	Hardware project	5.04	Charlton, Paul	Fast terminal program	Software review	6.29
	Techo time	Hardware project	6.07	VA, USA			
	Techo time	Hardware project	7.05	Christensen, Colin	HardMaster	Software review	9.25
	Techo time	Hardware project	8.05	Brisbane, Qld	Emulate file pointer	Software hints	9.22
	Techo time	Hardware project	9.05	Christensen, Garry	Hard disk drive - guided tour	Software hints	9.27
	Techo time	Hardware project	10.05	Brisbane, Qld	Quick hard disk back-up	Software hints	9.21
	Techo time	Hardware project	11.07				
	Techo time	Hardware project	12.07				
Anderson, Shane	Letter to editor	Merry Christmas	2.16	Demming, Bob	Program to type in	Grade calculator	1.15
Anonymous	Letter to editor	Participation	4.04	NY, USA			
	Letter to editor	Regional groups	4.04				



Author/Source	Title	Description	Page No.	Author/Source	Title	Description	Page No.
Dodd,Mike CA,USA	Trojan horse	General interest	3.27	Phillips,Terry	TI-Base tutorial	Data base	5.24
Gleed,Peter Melbourne,Vic	TIMES software competition	General interest	5.02		TisHUG software column	Club software	2.04
Hansen,Tor Ontario,Canada	TI-Writer, Funnelweb tutorial	Word processing	6.06		TisHUG software column	Club software	3.04
Harris,D.N	Conversion tables	Software hints	9.33		TisHUG software column	Club software	4.03
	Modems	General interest	5.33		TisHUG software column	Club software	5.05
	Plotic	General interest	12.20		TisHUG software column	Club software	6.03
Hope,Greg	Myarc, Geneve update	Hardware news	3.03		TisHUG software column	Club software	7.04
Ian TIBBS,USA	Deadline by Infocom	Adventure solution	2.07		TisHUG software column	Club software	9.04
	Infidel by Infocom	Adventure solution	2.07		TisHUG software column	Club software	10.03
					TisHUG software column	Club software	11.03
					TisHUG software column	Club software	12.03
Jingozian,Philip F MI,USA	Program to type in	Trigonometry	4.15	Rebel,Eric-Paul Netherlands	Transfers, cassette and disk	Software hints	9.32
Judd,Stephen	Games information	General interest	2.19	Relyea,Robert	Grade standardizer using X/B	Software hints	5.27
	Games information	Zork 1	5.17		Standard marks with Multiplan	Software hints	5.27
	Games information	Zork 2	6.15		Cassette finder review	Software review	6.08
Kanitz,Werner	From the trenches	General interest	10.30	Revitt,Steve Robinson,Adrian CA,USA	Calendar programs	General interest	12.21
Lang,Chris MD,USA	Game of Wit	Software review	10.16	Ryan,John	Letter to editor	New directors	2.03
Livingstone,E WA,USA	Program to type in	Statistical parameters	11.17		Letter to editor	Views of TisHUG	4.04
Magoo,Mr	Magoo's mumbblings	Software hints	5.21	Saunders,Larry	Adventure games	General interest	6.19
Maker,Vincent	Younger set	Program, adventure	4.06		Beyond video chess	Software review	6.12
	Younger set	Program, adventure	11.33		Buckaroo Banzai	Adventure hints	5.13
	Younger set	Program, adventure	12.18		Buckaroo Banzai	Adventure hints	6.24
	Younger set	Programs	2.09		Configuring PrEditor	Software review	3.19
	Younger set	Programs	3.28		Expanding your TI99/4A	Hardware review	4.11
	Younger set	Programs	7.09		Legends	Adventure hints	4.13
	Younger set	Programs	8.22		Legends	Adventure hints	6.21
	Younger set	Programs	10.33		Legends	Adventure hints	8.21
McGovern,Tony	Extended BASIC tutorial	Software hints	9.09		New TI99/4A	General interest	8.07
	Extended BASIC tutorial	Software hints	10.07		New programs and games	Software review	12.04
	Extended BASIC tutorial	Software hints	11.13		p-GRAM card and Geneve	Software review	5.25
	Extended BASIC tutorial	Software hints	12.05		PrEditor	Software review	3.17
McGovern,Will	Disk Hacker documentation	Software review	2.10		Press	Software review	5.08
Mudie,Ross	32K memory expansion	Hardware project	12.09		Public domain releases	Software review	4.02
	Assembler tutorial weekend	General interest	5.07		Questprobe with spiderman	Adventure hints	3.13
	Assembler tutorial weekend	General interest	6.05		Setup for smart modem	General interest	10.33
	Assembler tutorial weekend	General interest	8.02		TI-Artist upgrade	General interest	5.06
	Communicators	BBS information	2.05		Typewriter 99	Software review	5.11
	Communicators	BBS information	3.05	Scheidemantle,Paul OH,USA	Tiger does not sleep tonight	General interest	6.13
	Communicators	BBS information	7.11	Schreiber,Rolf	Getting your money's worth?	General interest	5.04
	Communicators	BBS information	11.05		Myarc rumours	General interest	5.26
	Communicators extra	BBS information	11.05		RAMdisk, another view	General interest	6.10
	Controlling a train set	Hardware project	2.31		AT music card	Hardware project	2.28
	Disassembling Extended BASIC	Software hints	4.22	Schubert,Peter Shafer,Mark KY,USA	Sector sharing	Software hints	6.09
	Disassembling X/B correction	Software correction	5.10	Shaw,Stephen England	Alternative micro show	General interest	1.15
	Handy hints	Hardware, software hint	5.32		Book reviews	General interest	6.12
	Joytalk, low cost printer port	Hardware review	3.24		Extended BASIC subprograms	Software hints	10.15
	Joytalk, low cost printer port	Software hints	4.10		MiniMemory meandering memories	Software hints	7.08
	Lego, not Logo turtle	Hardware project	6.31		Programming hint: not BASIC	Software hints	6.20
	Letter to editor	Life membership	3.03		Programs from TI+MES library	Software review	3.30
	Letter to editor	Modems	6.05		Programs from TI+MES library	Software review	6.34
	Sending mail on TEXPAC BBS	Software review	4.05		Programs from TI+MES library	Software review	7.11
	Shahzada endurance test	General interest	11.09		Rambles from UK	General interest	2.25
	TEXPAC BBS version 4F	BBS information	12.04		Rambles from UK	General interest	6.32
	Wire I/O update	Hardware project	3.24		Rambles from UK	General interest	11.12
Murrell,William USA	Glossarial on acronyms	General interest	11.06		Rambles from UK	General interest	12.29
Nollan,Joe WA,USA	Working with numbers	Software hints	12.28		Starting using TI99/4A	Software hints	12.17
Olson,Stuart IL,USA	Mass transfer	Software review	7.12		TI Planner, second thoughts	Software hints	2.24
Paine,John	Anatomy of a pirate	General interest	7.30		TI Runner level editor	Software review	7.17
	Death of a computer	General interest	3.07		Turbo pasc 99 tutorial	Software review	5.31
	Hard disk drive data	Hardware review	9.23	Sheehan,Craig	c99 tutorial	Software hints	8.14
	Letter to editor	Magic squares	6.03		c99 tutorial	Software hints	9.12
Peterson,Jim OH,USA	Printer problems	Software hints	11.31		c99 tutorial	Software hints	10.12
	TI-Writer bug	Word processing	3.26		c99 tutorial	Software hints	12.23
	Tigercub public domain catalog	Software review	5.18		Extended display package	Software review	2.07
	Tigercub public domain catalog	Software review	5.19		Extended display package	Software review	3.06
	Tigercub public domain catalog	Software review	8.17		Extended display package	Software review	4.07
	Tigercub public domain catalog	Software review	11.25		Extended display package	Software review	5.09
	Tigercub software catalog	Software review	4.27		Full day workshop	Club news	6.04
	Tigercub software catalog	Software review	6.17		Full day workshop	Club news	11.04
	Tigercub software catalog	Software review	8.19		Strings in Extended BASIC	Software hints	8.27
	Tigercub software catalog	Software review	11.15	Smoley,Martin OH,USA	TI-Base tutorial	Data base	8.11
	Tigercub tips index	Software index	6.16		TI-Base tutorial	Data base	9.13
	Tips from the tigercub #29	Software hints	4.25		TI-Base tutorial	Data base	10.09
	Tips from the tigercub #30	Software hints	5.15		TI-Base tutorial	Data base	11.21
	Tips from the tigercub #31	Software hints	6.25		TI-Base tutorial	Data base	12.25
	Tips from the tigercub #32	Software hints	7.15	Stanford,Jeff USA	Forth dimension	Software hints	12.27
	Tips from the tigercub #33	Software hints	8.25	Sumner,Mark England	Program to type in	Wankapillar	12.15
	Tips from the tigercub #34	Software hints	9.19	Swedlow,Jim CA,USA	Disk utilities, X/B tips	Software review, hints	11.23
	Tips from the tigercub #35	Software hints	10.13		Extended BASIC tips	Software hints	6.10
	Tips from the tigercub #36	Software hints	11.19		Funnelweb configuration	Software hints	2.13
	Tips from the tigercub #37	Software hints	12.19		TI bits 6, 7, 10, 11	Software hints	2.15
	Tips from the tigercub #53	Software hints	2.29	Takach,Ben	Answer is abc, question?	General interest	10.03
	Tips from the tigercub #54	Software hints	3.25		Backing up the hard disk	General interest	9.22
	Tips from the tigercub #55	Software hints	5.23		Clock programs with hard disk	Software hints	8.15
	Tips from the tigercub #56	Software hints	6.27		Mechatronic 80 column card	Hardware review	8.31
	Tips from the tigercub #57	Software hints	7.19		PE box unregulated voltages	Hardware project	3.22
	Tips from the tigercub #58	Software hints	10.21		TisHUG in 1989	General interest	3.02
Phillips,Terry	Agenda for AGM, 1989	Club news	1.01	Telford,Darren	Trials with Myarc DM5	General interest	10.28
	Minutes of 1988 AGM	Club news	1.02	Traver,Barry PA,USA	What is GIF?	General interest	5.30
	Secretary's notebook	Club news	2.02	Trott,Geoff	XJB documentation	Software review	11.30
	Secretary's notebook	Club news	3.03		Bug report	DM1000 version 3.5	4.06
	Secretary's notebook	Club news	4.03		Bug report	Myarc hard controller	9.04
	Secretary's notebook	Club news	5.03		Bug report	TI-Writer and formatter	5.29
	Secretary's notebook	Club news	6.03		c99 printer setup	Software hints	6.33
	Secretary's notebook	Club news	7.03		Fairware author of month	Ottawa user group	8.03
	Secretary's notebook	Club news	8.03		Fairware author of month	Ottawa user group	9.02
	Secretary's notebook	Club news	9.03		Fairware author of month	Stuart Olsen	10.02
	Secretary's notebook	Club news	10.03		Fairware author of month	Stuart Olsen	11.03
	Secretary's notebook	Club news	11.03		Fairware author of month	Tony McGovern	7.20
	Secretary's notebook	Club news	12.03		Interfacing to RS232 PIO port	Hardware project	4.09
					Making your own dictionaries	Software hints	8.23
					Making your own dictionaries	Software hints	9.30
					Making your own dictionaries	Software hints	10.31
					Printing 132 columns with ATMFPC	Software hints	5.12
					RAMdisk power supply problems	Hardware project	9.06
					Reliable LOAD interrupt	Hardware project	3.21
					Starting a hard disk	Hardware review	4.19

Author/Source	Title	Description	Page No.	Author/Source	Title	Description	Page No.
Trott, Geoff	They're off	General interest	2.01	Author unknown	Forth to you too!	Software hints	9.33
	They're off	General interest	3.01	USA	Forth to you too!	Software hints	10.23
	They're off	General interest	4.01	USA	Forth to you too!	Software hints	11.27
	They're off	General interest	5.01	USA	Forth to you too!	Software hints	12.27
	They're off	General interest	6.01		From the bulletin board	Mail to all	2.05
	They're off	General interest	7.01		From the bulletin board	Mail to all	3.05
	They're off	General interest	8.01		From the bulletin board	Mail to all	4.06
	They're off	General interest	9.01		From the bulletin board	Mail to all	5.10
	They're off	General interest	10.01		From the bulletin board	Mail to all	6.05
	They're off	General interest	11.01		From the bulletin board	Mail to all	7.11
	They're off	General interest	12.01		From the bulletin board	Mail to all	9.31
	TiSHUG software awards	Club news	5.06		From the bulletin board	Mail to all	10.33
Verhey, Dwayne Ontario, Canada	TI-Writer, Funnelweb tutorial	Word processing	8.06		From the bulletin board	Mail to all	11.05
Warburton, Dick	Co-ordinators report	General news	3.02	USA	Graphics interchange format	Software hints	7.31
	Co-ordinators report	General news	4.02		Problem solving procedures	General interest	8.24
	Co-ordinators report	General news	5.02		Program to type in	Animal learner	10.18
	Co-ordinators report	General news	6.02		Program to type in	Artist board	2.23
	Co-ordinators report	General news	7.02	99'er, USA	Program to type in	Brain teaser	2.20
	Co-ordinators report	General news	8.02		Program to type in	County fair derby	2.21
	Co-ordinators report	General news	9.02		Program to type in	Kingdom of Sumaria	10.17
	Co-ordinators report	General news	10.02		Program to type in	Magic squares	3.14
	Co-ordinators report	General news	11.02		Program to type in	Miniature golf	3.15
	Co-ordinators report	General news	12.02	USA	Regression analysis	General interest	4.14
Welham, Warren	Publications library report	Club library	3.27		Project Management	General interest	12.14
	Publications library report	Club library	5.08		Regional group reports	General interest	2.43
Wilkenson, Derek	Interface IBM style keyboard	Hardware project	9.05		Regional group reports	General interest	3.31
Winkler, Lutz USA	Forth tidbits 1 to 6	Software hints	10.23		Regional group reports	General interest	4.31
	Forth tidbits 7, 8 and 9	Software hints	11.27		Regional group reports	General interest	5.35
Author unknown USA	Advanced Diagnostics review	Software review	8.29		Regional group reports	General interest	6.35
	Author index volume 7, 1988	General interest	1.08		Regional group reports	General interest	7.35
	c99 quick reference	Software hints	12.24		Regional group reports	General interest	8.35
	Electronic Bible	General interest	6.05	USA	Regional group reports	General interest	9.35
	Extended BASIC screen colours	Software hints	12.12		Regional group reports	General interest	10.35
					Regional group reports	General interest	11.35
					Regional group reports	General interest	12.31
					Ten commandments for technician	General interest	11.08
					Title index volume 7, 1988	General interest	1.03

## TI-Base version 2.0

by Rick Lilley, from Melbourne TIMES

Wow! What a data base! Now we are cooking. I am, as you may have gathered, impressed with TI-Base! The newsletter deadline is getting real close so I will have to be brief. Version 1 was good, but version 2 is much better, not perfect, but better. I will not waste a lot of time describing the merits of the program, I am sure most of you have read about it by now.

I think this TI-Base is the greatest invention since the discovery of the round ball, and I am going to show you what I have been learning to do with it.

When you first boot this data base you will see a date prompt, and after that a "STATUS" display, this is automatically put on the screen to show you all the defaults in effect. To change anything, such as which disk drive you are going to use for your data disk ("DATDISK"), you simply type "SET DATDISK DSK2" or whatever, (and of course enter) the period is not required. Then type "DISPLAY STATUS" and you will see the status display again with changes. To set this permanently (for rebooting) you will have to type "SET DATDISK DSK?" the ? being whatever drive your program disk is in, and then type "MODIFY COMMAND SETUP". You will soon see this command file in front of you. Use the insert line feature, FNCT[4], and type "SET DATDISK DSK?", this time the ? being the drive you want to use for your files, Drive 2, or whatever. Then type "DO SETUP", and SETUP will run and display STATUS again with your changes displayed.

You can boot your program disk ("PRGDISK") from any drive and it will automatically be listed on the STATUS as being from that drive. As Colonel North said, "I think it is neat!"

I have been setting up a data base named "LOG" for my work log. I am including here for all to see, the file structure, the command file, "MENU" I have built, and the command file called "DISP". This is used to show my files in an orderly fashion from any letter in the alphabet I choose, (sorted alphabetically by name of course). The command file "PRINT" handles my main print out functions for now, it may have some additions later. Also the file structure.

I might mention here that you can sort your files by more than one field at a time (up to 8), but only the first field will be searched on a "FIND" directive. By the way, the manual states that you can load from three

different cartridges. I have to use the MiniMemory cartridge or else I get the "CAN'T GET DYNAMIC BUFFER SPACE" message on the screen. I suppose I could get around this problem by breaking up my command files into smaller modules with more calls, but since I have the cartridge I might as well use it, (the MiniMemory gives you almost 5000 extra bytes of "DYNAMIC MEMORY"). The return to your original command file is automatic from a called file just like a "GOSUB" in BASIC. The menu I plagiarized somewhat, from the menu that comes with the tutor disk.

The program, presumably programmed by a genius, is not yet well documented, although version 2 has more information in its manual than version 1 did. And I made a discovery! Nosing about in the library I discovered a book about dBASE III. "Using dBASE III" by Edward Jones. It is not hard to see where our "RELATIONAL" data base comes from. It is amazing how many of the functions and commands are identical to those of TI-Base. I learned a great deal about our D/BASE from this book. (Hint: Go find one like it.)

Incidentally, the reason I was able to "see" the command file from the tutor disk is because one of the improvements to version 2 is the "LIST" function which allows you to list any command file, that is, any file with "/C" on the end of its name on your disk catalogue. To list file structures as I have, to put in the newsletter, you have to use the "SNAP" command, which allows you to dump whatever is on the screen to your printer. There always seems to be a way to do whatever you want to do with this data base. Finding out how to do it is the only problem! It is very powerful.

So far I have been able to convert my old "CFS" data base of 150 records over to this one and even to combine two similar fields into one larger field. It was not easy, but it worked, (finally). More on that in another article. I have not yet begun to explore the math functions, or figured out how to print labels for a name and address file I am working on. This data base has the ability to save to disk in DV80 format so I am sure mail merge is a possibility too.

I am going to have to leave you now, with the promise of more to come, (if I do not get shot for spending so much time at the keyboard, (there have been some rumblings))! I hope you all get some benefit from my files (a picture is worth a thousand words etc.), and I hope any of you TI99/4Aer's out there who have some more to contribute about this data base will share it with us. In this way I am sure we will all learn more about it. (The grapevine!) God knows, I could use some help!

```

TOP
  REPLACE TEMP WITH " E " | .DATE. ;
  | " ";
  | " ** NorthCoast Roster **"
  PRINT TEMP
  PRINT BLNK
DO DSK2.RSTR2WH
  PRINT BLNK
  REPLACE TEMP WITH " Total " ;
  | "Membership is:" | COUNT
  PRINT TEMP
CLEAR
  DO DSK2.FIN1
RETURN
*
* PRSTR2          Save as PRSTR2/C
* *****      Print Roster  12/31/88

```

```

E WHILE .NOT. (EOF)
  WHILE (XP < CUTOFF) .AND. ;
    (.NOT. (EOF))
  MOVE
  ENDWHILE
  IF (EOF)
    RETURN
  ENDF
  IF XP = PRDT
    REPLACE TEMP WITH " " ;
    "___" | FN | LN | PH | XP | ;
    " Please Renew"
  PRINT TEMP
  ENDF
  IF (XP < PRDT) .AND. (XP <> CUTOFF)
    REPLACE TEMP WITH "Pay Your Dues " | ;
    "___" | FN | LN | PH | XP
  PRINT TEMP
  ENDF
  IF XP = CUTOFF
    REPLACE TEMP WITH "Last Chance! " | ;
    "___" | FN | LN | PH | XP | ;
    "<= Pay Up!"
  PRINT TEMP
  ENDF
  IF XP > PRDT
    REPLACE TEMP WITH " " | ;
    "___" | FN | LN | PH | XP
  PRINT TEMP
  ENDF
  MOVE
  REPLACE COUNT WITH COUNT + 1
  ENDWHILE
RETURN
*
* RSTR2WH Save as RSTR2WH/C
* ***** Print Roster While 01/02/89

```

[illegible]

## Tips from the Tigercub #38

by Jim Peterson, Tigercub Software, USA

Copyright 1986 TIGERCUB SOFTWARE  
156 Collingwood Ave., Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in BASIC and Extended BASIC, available on cassette or disk, only \$3.00 each plus \$1.50 per order for ppm. Entertainment, education, programmer's utilities.

Descriptive catalog \$1.00, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter number 1 through 14, 50 original programs and files, just \$15 postpaid.

Tips from the Tigercub volume 2, another disk full, complete contents of numbers 15 through 24, over 60 files and programs, also just \$15 pp.

\*\*\*\*\*  
\* Tips from the Tigercub volume 3 is now ready. \*  
\* Another 62 programs, routines, tips, tricks, \*  
\* numbers 25 through 32. Also \$15 pp. Any two \*  
\* Tips disks \$27 or all 3 for \$35 pp. \*  
\*\*\*\*\*

Nuts Bolts (No. 1), a full disk of 100 Extended BASIC utility sub-programs in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using sub-programs, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Nuts Bolts No. 2, another full disk of 108 utility sub-programs in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST	PROGRAM-TUTOR
PROGRAMMER'S UTILITIES	KID'S GAMES
BRAIN GAMES	BRAIN TEASERS
BRAIN BUSTERS!	MORE GAMES
ACTION REFLEX AND CONCENTRATION	MANEUVERING GAMES
TWO-PLAYER GAMES	WORD GAMES
ELEMENTARY MATH	MIDDLE/HIGH SCHOOL MATH
MUSICAL EDUCATION	VOCABULARY AND READING
KALEIDOSCOPIES AND DISPLAYS	

For descriptions of these send a dollar for my catalog!

I have discovered a rare bug in the 28-Column Converter, published in Tips #18, which will cause an I/O 25 ERROR if the very last line of the program being converted happens to have exactly 80 characters. You can fix it by adding a line:

```
215 IF EOF(1)=1 THEN 260
```

There is also a rare bug in the SIDEWAYS subroutine on my Nuts Bolts #2 disk, which prevents turning some redefined character sets sideways. If you are one of those who BOUGHT that disk from me, you can fix it by changing the L=LEN(B\$) in line 21639 to L=64.

I was in too much of a hurry to go fishing when I put the last couple of Tips together. In the Gordian Knot in Tips #35, I left out some essential instructions. Please add:

```
131 DISPLAY AT(11,1):" When you cross your  
track,": "press 0 to go over, U to go": "under, C to go  
across."
```

To make that fit, you will have to change the DISPLAY AT in line 130 to (8,1), in line 140 to (15,1) and in line 150 to (20,1), also the ACCEPT At in 160 to (20,11). And this change will prevent a lockup when you reach a border:

```
200 D=D-1 :: IF ABS(D-D2)=2 OR R+(D=1)=0 OR R-(D=3)=25  
OR C+(D=4)=2 OR C-(D=2)=31 THEN 180 :: GOSUB 510 :: IF  
D<>D2 THEN GOSUB 450
```

I wrote the dulcimer music in Tips #36 in BASIC, but I forgot to test it in BASIC. It actually runs much better in Extended BASIC, but will run fairly well in BASIC if you delete the delays in lines 280 and 300.

If you liked the ESCHER ART in Tips #37, these modifications will improve it considerably:

```
110 DISPLAY AT(12,1):"Press-": " Q for new pattern":  
" B to change background": " F to change foreground":  
" R to reverse colours "::::"Any key to start"  
280 A=INT(6*RND+3):: H=INT(24/A) :: RX=24-H*A ::  
HC=INT(28/A):: CX=28-HC*A :: W=ABS(HC/2=INT(HC/2))  
-(RX>0):: DIM M(8,8):: FOR P=1 TO A  
330 IF K<>66 THEN 346  
340 BC=BC+1+(BC=16) :: IF BC=F THEN 340 ELSE 347  
346 IF K<>70 THEN 360 :: F=F+1+(F=16) :: IF F=BC  
THEN 346  
347 FOR S=7 TO 14 :: CALL COLOR(S,F,BC):: NEXT S ::  
GOTO 310  
350 ! **DELETED LINE **  
360 IF K<>ASC("R")THEN 310 :: T=F :: F=BC :: BC=T ::  
GOTO 347  
600 GOSUB 900 :: FOR T=1 TO A :: DISPLAY AT(R-1+T,C):  
M$(V,T):: NEXT T :: NEXT C  
601 IF CX>0 THEN AA=A :: GOSUB 800  
605 GOSUB 1000 :: NEXT R  
606 IF RX=0 THEN 610  
607 GOSUB 1000 :: FOR C=1 TO A*HC STEP A :: GOSUB 900  
:: FOR T=1 TO RX :: DISPLAY AT(R-1+T,C):M$(V,T) ::  
NEXT T :: NEXT C  
608 IF CX>0 THEN AA=RX :: GOSUB 800  
800 GOSUB 900 :: FOR T=1 TO AA :: DISPLAY AT  
(R-1+T,C):SEG$(M$(V,T),1,CX):: NEXT T :: RETURN  
900 V=V+1+(V=4)*4 :: RETURN  
1000 V=V+W :: V=V+(V>4)*4 :: RETURN
```

I had a letter from a teacher who was using the PRK module to keep student grades, and wanted to know how to average them. It can be done, but is so impractical that I wrote this program. While I was at it, I speeded up the loading and saving to cassette greatly by converting the grades to an ASCII string and combine the student's name and all grades into one record.

```
100 DIM N$(50),T(50,20)  
110 CALL CLEAR  
120 PRINT " TEACHER'S HELPER": : : :  
130 REM - by Jim Peterson  
140 PRINT "(1)CREATE A FILE?": "(2)ADD TO FILE?": "  
(3)LOAD A FILE?": "(4)SAVE A FILE?": "(5)PRINT A  
FILE?"  
150 PRINT "(6)CORRECT A FILE?": "(7)COMPUTE AVERAGES?":  
"(8)QUIT?"  
160 CALL KEY(0,K,S)  
170 IF (S=0)+(K<49)+(K>56)THEN 160  
180 ON K-48 GOTO 190,250,610,800,380,990,1120,1510  
190 X=0  
200 INPUT "SUBJECT? ":S$  
210 GOSUB 1370  
220 INPUT "TEST #? ":N  
230 GOSUB 1440  
240 GOTO 140  
250 PRINT :: "(1)ADD NAMES?": "(2)ADD GRADES?"  
260 CALL KEY(0,K,S)  
270 IF (S=0)+(K<49)+(K>50)THEN 260  
280 ON K-48 GOTO 290,310  
290 GOSUB 1370  
300 GOTO 140  
310 INPUT "TEST #? ":Q  
320 IF T(1,Q)=0 THEN 350  
330 PRINT :: "TEST #":STR$(Q); " ALREADY RECORDED"  
340 GOTO 140  
350 N=Q  
360 GOSUB 1440  
370 GOTO 140  
380 CALL CLEAR
```



```

390 PRINT "OUTPUT TO":"(1)SCREEN?":"(2)PRINTER?"
400 CALL KEY(0,K,S)
410 IF (S=0)+(K<49)+(K>50)THEN 400
420 IF K=49 THEN 460
430 INPUT "PRINTER DESIGNATION? ":P$
440 OPEN #2:P$
450 F@=2
460 PRINT "PRESS ANY KEY TO PAUSE": :
470 PRINT #F@:S$: :
480 FOR J=1 TO X
490 PRINT #F@:"":N$(J)&" ";TAB(10);
500 FOR K=1 TO HN
510 PRINT #F@:T(J,K);
520 NEXT K
530 CALL KEY(0,K,S)
540 IF S<>0 THEN 530
550 NEXT J
560 PRINT #F@
570 IF F@=0 THEN 140
580 F@=0
590 CLOSE #2
600 GOTO 140
610 PRINT :;:"(1)CASSETTE?":"(2)DISK?"
620 CALL KEY(0,K,S)
630 IF (S=0)+(K<49)+(K>50)THEN 620
640 ON K-48 GOTO 650,670
650 OPEN #2:"CS1",INPUT ,FIXED
660 GOTO 690
670 INPUT "FILENAME? DSK":F$
680 OPEN #2:"DSK"&F$,INPUT
690 INPUT #2:X,HN,S$
700 FOR J=1 TO X
710 INPUT #2:K$
720 N$(J)=SEG$(K$,1,POS(K$,CHR$(255),1)-1)
730 K$=SEG$(K$,POS(K$,CHR$(255),1)+1,255)
740 FOR K=1 TO HN
750 T(J,K)=ASC(SEG$(K$,K,1))-50
760 NEXT K
770 NEXT J
780 CLOSE #2
790 GOTO 140
800 PRINT :;:"(1)CASSETTE?":"(2)DISK?"
810 CALL KEY(0,K,S)
820 IF (S=0)+(K<49)+(K>50)THEN 810
830 ON K-48 GOTO 840,860
840 OPEN #2:"CS1",OUTPUT,FIXED
850 GOTO 880
860 INPUT "FILENAME? DSK":F$
870 OPEN #2:"DSK"&F$,OUTPUT
880 PRINT #2:X:HN:S$
890 FOR J=1 TO X
900 K$=""
910 FOR K=1 TO HN
920 K$=K$&CHR$(T(J,K)+50)
930 NEXT K
940 PRINT #2:N$(J)&CHR$(255)&K$
950 K$=""
960 NEXT J
970 CLOSE #2
980 GOTO 140
990 CALL CLEAR
1000 INPUT "STUDENT'S NAME?":Q$
1010 FOR J=1 TO X
1020 IF N$(J)=Q$ THEN 1060
1030 NEXT J
1040 PRINT :;:"NAME NOT FOUND": :
1050 GOTO 140
1060 INPUT "CORRECT WHICH TEST? (0 TO QUIT) ":C
1070 IF C=0 THEN 1110
1080 PRINT :;:N$(J);"S TEST #";STR$(T(J,C)): :
1090 INPUT "CORRECT TO? ":T(J,C)
1100 GOTO 1060
1110 GOTO 140
1120 CALL CLEAR
1130 PRINT "OUTPUT TO":"(1)SCREEN?":"(2)PRINTER?"
1140 CALL KEY(0,K,S)
1150 IF (S=0)+(K<49)+(K>50)THEN 1140
1160 IF K=49 THEN 1200
1170 INPUT "PRINTER DESIGNATION? ":P$
1180 OPEN #2:P$
1190 F@=2
1200 PRINT #F@:S$
1210 FOR J=1 TO X
1220 PRINT #F@:N$(J);" AVERAGE ";

```

```

1230 FOR K=1 TO HN
1240 TT=TT+T(J,K)
1250 NEXT K
1260 AV=TT/HN
1270 TAV=TAV+AV
1280 PRINT #F@:AV
1290 TT=0
1300 NEXT J
1310 PRINT #F@:"CLASS AVERAGE ";TAV/X
1320 TAV=0
1330 IF F@=0 THEN 1360
1340 F@=0
1350 CLOSE #2
1360 GOTO 140
1370 PRINT :;:"STUDENT'S NAMES - ":"type END when
finished": :
1380 X=X+1
1390 M$="NAME #"&STR$(X)&" "
1400 INPUT M$:N$(X)
1410 IF N$(X)<>"END" THEN 1380
1420 X=X-1
1430 RETURN
1440 FOR J=1 TO X
1450 M$=N$(J)&"'S GRADE? "
1460 INPUT M$:T(J,N)
1470 NEXT J
1480 IF N<HN THEN 240
1490 HN=N
1500 RETURN
1510 END

```

The reason that 50 is added to the value in line 920, before saving, and subtracted again in line 750 after loading, is because of a quirk of the computer that I do not recall seeing in print anywhere. Did you know that INPUT will read a string beginning with ASCII 0, 2, 4, 7, 10, 12, 14, 18, 20, 26, 27, 31, 32, or 44 as a null string (a blank), and will drop these characters at the end of a string? And ASCII 32 will be dropped at the beginning or end of a string. And ASCII 0 within a string, or ASCII 34 anywhere, will crash, while ASCII 44 within a string will lose the rest of the string. I should have known what ASCII 0, 32 (the space), 34 (quotes) and 44 (comma) would do, but why the others? LINPUT will accept anything, of course, but I wanted to keep this in BASIC for the teachers who are struggling along without the Extended BASIC module or disk drive. Chick De Marti published in LA 99ers TOPICS the surprising discovery that PRINT USING and DISPLAY USING can read the IMAGE format from a variable, array or string! Which led me to some fooling around:

```

100 !PRINT USING DEMO by Jim Peterson, based on a
discovery by Chick Marti
110 CALL CLEAR :: RANDOMIZE :: CALL SCREEN(5):: FOR
S=2 TO 14 :: CALL COLOR(S,S,S) :: NEXT S
120 N=INT(13*RND+1):: C$=CHR$(8*N+32-(N=4))
130 FOR J=N TO 12 :: A$=RPT$(" ",J)&"#&RPT$(" ",26-
J*2)&"#" :: PRINT USING A$:C$,C$ :: NEXT J
140 FOR J=12 TO N STEP -1 :: A$=RPT$(" ",J)&"#&RPT$
(" ",26-J*2)&"#" :: PRINT USING A$:C$:C$ :: NEXT J
:: GOTO 120

```

Here is one last Tigercub challenge. What is the longest possible one-liner? And what is the longest possible one liner that actually does something?

MEMORY FULL Jim Peterson

## For Sale

The following items are surplus to my requirements. They all come in their original boxes and with all documentation:

TI99/4A beige console	\$50
TI Speech Synthesizer	\$50
TI Extended BASIC	\$40
Shuttle Modem (300 baud only)	\$40

### Wanted

PIO adapter for Peter Schubert RS232/Modem unit.  
Contact Jim Ellis on (064)92 1772 after hours.

# TI-Writer version 4.0

by Robert Green, Canada

Version 4.0 is a major new Fairware release of TI-Writer. The modifications have been extensive in order to remove many of the annoyances of the original package. At the same time, the new version is strictly compatible with the original.

Fairware contributions of \$10 can be sent to:  
RAG SOFTWARE, R.A. Green, 1032 Chantenay Drive,  
Gloucester, Ontario CANADA K1C 2K9

## Editor Improvements.

1. The editor runs independent from the cartridge (or with it if you like).
2. The performance (i.e. speed) of all features has been improved, with special attention to some features.
3. A dramatic improvement in the speed of move, copy and delete lines. Move lines is instantaneous and will never give the "out of memory" condition. Delete lines, in most cases, is also nearly instantaneous.
4. All cursor movement from line to line has been speeded up when no lines are changed.
5. All keyboard input has been speeded up in an attempt to prevent loss of characters when "wrapping" to the next line.
6. A new command, QQ, has been added to exit immediately from the editor without further prompting.
7. A number entered as a command is equivalent to Show line, that is, the line whose number is entered is positioned at the top of the screen.
8. In command mode, simply pressing <ENTER> will return you to edit mode.
9. Two new control codes have been added. CTRL[,] positions to the top of the file (i.e. Show line 1). CTRL[.] positions to the last line of the file (i.e. Show line E).
10. The CHARA1 file is no longer required to define a new character set for the editor.
11. An install or configuration program is provided that allows you to tailor the editor for your environment. You can:
  - a. define your printer for PF,
  - b. set the initial screen colours,
  - c. set the initial tabs,
  - d. set word wrap initially on or off,
  - e. set display of line numbers initially on or off,
  - f. define the character set.

## Editor Notes.

1. No features of the Editor have been removed, although SD is different as it was done by code in the cartridge GROM.
2. The text buffer is exactly the same size.
3. As mentioned in the TI-Writer manual but not stressed enough, it is important to immediately do a save file and then reload it after a Recover Edit.

## Formatter Improvements.

1. The Formatter now runs independent from the cartridge (or with it if you like).
2. The overall performance (i.e. speed) of the Formatter has been improved.
3. There has been a dramatic reduction in the size of the Formatter on disk which makes for faster loading.

4. The Formatter's "format commands" can now be entered in upper, lower or mixed case.
5. When multiple format commands are used on a single line (separated by semi-colons) a period can precede all commands.
6. The bug in handling asterisks when not in mailing list mode and the bug in handling double at signs or double ampersands in centred lines have been fixed. Try your existing Formatter on the file BUGDEMO.
7. Eight new format commands have been added to the Formatter. They are described below.
8. A new response to the prompt, "Pause at end of page?", is provided to make using letterhead or single sheet paper easier. The response "L" indicates that the formatter is to pause at the end of each page and that the normal spacing at the top of the page is not to be done so that letterhead paper can be positioned in the printer at the correct place for the first line of text to be printed. The formatter normally prints two blank lines, the HE line and another blank line at the top of pages. None of these will be printed when the "L" response is used, but the line number will still be set to five.
9. The disk number in the filenames for IF, ML and CH format commands may be specified as an asterisk to indicate the same drive as the main input file. This allows a document to be printed independent of the drive selected by the user.
10. An install or configuration program is provided that allows you to define your printer for the formatter.

## Formatter Notes.

No features have been removed.

## Installing Version 4.0

First, make a working copy of the distribution disk. If you have a character set module that you want installed into the Editor then it must be made available on a separate drive or must be copied onto the working disk. You will be asked for the filename of the character set module during installation.

The two installation programs, EDITINST and FORMINST, can be loaded and run using: Editor Assembler Option 5, TI-Writer Option 3, or Extended BASIC with program INLOAD.

The installation programs prompt for their inputs.

## Loading Version 4.0

The new Editor and Formatter can be loaded in a variety of ways. Both programs are now standard "Editor Assembler Option 5" programs. Of course, they can be loaded by the TI-Writer cartridge in the normal way, or by using Option 3. The Editor is completely independent and loads all VDP registers and tables. The Formatter requires the VDP set up as for Editor Assembler Option 5. The following special loaders are provided to simulate the TI-Writer Cartridge Menu for loading the Editor and Formatter.

1. Extended BASIC "LOAD" program.

This program can be easily modified to add other programs to the menu. The first DATA statement has the menu heading and the number of menu items. The other DATA statements have the program environment type, the program name and the menu text for each program on the menu. Each item is in the form:

"Tnnnnnnnnnn tttttt..."

where "T" is the environment type the program requires, and is either "E" for Editor Assembler environment or "X" for Extended BASIC environment. "nnnnnnnnnn" is the

name of the program file to be loaded and must be exactly 10 characters, padded if necessary on the right with blanks. "tttttt" is the text for the menu display.

The disk number from which the programs are loaded can also be changed by changing statement 7:

```
CALL LOAD(-123,49)
```

where 49 is ASCII "I". Note: Do not change statements 1 or 2 as the loader is hidden between them.

2. Editor Assembler Supercart menu program.

When program "TIWSEA" is loaded via Option 5 into an Editor Assembler Supercart (i.e. RAM at >6000->7FFF) it presents a menu for TI-Writer Version 4.0. It also leaves itself as a ROM menu item for consoles that support ROM cartridges.

Note that the file names entered for Options 3, 4 and 5 are retained and are available for later use. The drive number from which the Editor and Formatter are loaded can be easily patched in TIWSEA either on disk or when loaded into the Editor Assembler Supercart.

3. Mini Memory resident loader.

When program "TIWMM" is loaded via Option 3 into the Mini Memory and run with program name "TI-Writer" it presents a menu for TI Writer Version 4.0. Once loaded, the program can be reused until something else is loaded into the Mini Memory.

Note that the file names entered for Options 3, 4 and 5 are retained and are available for later use. The drive number from which the Editor and Formatter are loaded can be easily patched in TIWMM using Easy Bug. Location >73D5 is the drive number for the Editor and location >73F1 is the drive number for the Formatter.

Distribution Disk Contents

BUGDEMO	Demonstrates two Formatter bugs
CHARA1	A true lower case character set
EDITA1	The Editor, segment 1
EDITA2	The Editor, segment 2
EDITINST	Installation program for the Editor
FORMA1	The Formatter
FORMINST	Installation program for the Formatter
INLOAD	Extended BASIC loader for the installation programs
LOAD	Extended BASIC loader for the Editor and/or Formatter
TIWMM	Mini Memory TI-Writer Menu object text, program name: TI-Writer
TIWSEA	E/A Supercart TI Writer Menu
TIWV40	This Version 4.0 writeup

New Format Commands.

Printer Control coded as:

```
PC n1,n2,n3,...
```

which causes the control codes n1,n2,n3,... to be sent directly to the printer without changing the line count. This can be used for printer setup without the trouble of TL. Note that like most format commands, PC causes a break in the text and thus cannot be used in the middle of a line.

Define Underscore Control Character coded as:

```
DU n
```

where n is the number of the code to be used as the underscore begin character. The initial setting for DU is 38, the ampersand.

Define Boldface control character coded as:

```
DB n
```

where n is the number of the code to be used as the boldface or overstrike begin character. The initial setting for DB is 64, the at sign.

Define Mailing List control character coded as:

```
DM n
```

where n is the number of the code to be used as the mailing list insert character. The initial setting for DM is 42, the asterisk.

Define Required Blank character coded as:

```
DR n
```

where n is the number of the code to be used as the required blank character. The initial setting for DR is 94, the caret.

As Is Text Begin coded as:

```
AI
```

This format command is similar to NF except that the left margin is still observed. As Is text is ended by an FI format command.

Conditional Page Eject coded as:

```
CP n
```

A page break will occur if there is less than n lines remaining on the current page; otherwise, the CP is ignored.

Chain Files coded as:

```
CF filename
```

When this command is encountered, the main input file is closed, and the user is prompted to insert the disk for the named file. When the user presses <ENTER>, the named file is processed. This allows changing of disks and the processing of an unlimited length document. Note this command is invalid in an "included" file (just as IF commands may not be nested). In addition, only a single copy of the document will be printed when chained files are used.

Reference Sheets

The complete list of commands and function codes for both the Editor and the Formatter is given below.

Editor Commands

CMD	FUNCTION	PARAMETERS
C	Copy lines	start stop after
D	Delete lines	start stop
DF	Delete file	filename
E	Edit mode	--
F	Files help	--
FS	Find string	[startcol] [endcol] /string/
L	Lines help	--
LF	Load file	[after][start][end] filename
M	Move lines	start stop after
P	Purge text	Y N
PF	Print file	[C][L][start][stop] printername
Q	Quit	E P S
QQ	Quick quit	--
RE	Recover Edit	Y N
RS	Replace String	[startcol][endcol] /old/new/
S	Show line	number
SD	Show Directory	disknumber
SF	Save File	[start][stop] filename
SH	Search help	--
T	Tabs	I T L R
n	Line n	--

FCTN	ACTION	CTRL	ACTION
1	Delete Character	1	Oops
2	Insert Character	2	Reformat
3	Delete Line	3	Screen Colours
4	Roll Down	4	Next Paragraph
5	Next Window	5	Duplicate Line
6	Roll Up	6	Last Paragraph
7	Tab	7	Word Tab
8	Insert Line	8	New Paragraph
9	Command Mode	9	New Page
0	Line Numbers Toggle	0	Word Wrap Toggle
=	Command Mode	A	Roll Down
E	Cursor Up	B	Roll Up
D	Cursor Right	C	Command Mode
S	Cursor Left	D	Cursor Right
X	Cursor Down	E	Cursor Up
		F	Delete Character
		G	Insert Character
		H	Last Paragraph
		I	Tab
		J	Next Paragraph
		K	Delete to End of Line
		L	Home Cursor
		M	New Paragraph
		N	Delete Line
		O	Insert Line
		P	New Page
		Q	-----
		R	Reformat
		S	Cursor Left
		T	Back Tab
		U	Special Character Mode
		V	Beginning of Line
		W	Word Tab
		X	Cursor Down
		Y	Left Margin Release
		Z	Oops
		comma	Show Line 1
		period	Show Line E

## Formatter Commands

CMD	FUNCTION
@	Begin Boldfaceing (See DB)
&	Begin Underscoring (See DU)
caret	Required Blank (See DR)
*	Mailing List Variable (See DM)
c/r	Break Text
p/a	Break Page
AD	Begin Right Margin Justification
AI	Begin As Is Text
BP	Break Page
BR	Break Text
CE n	Centre n Lines
CH filename	Chain Files
CO text	Comment
CP n	Conditional Page Break
DB n	Define Boldface Character
DM n	Define Mailing List Variable Char
DP n:text	Define Mailing List Prompt
DR n	Define Required Blank Character
DU n	Define Underscore Character
FI	Begin Text Filling
FO text [%]	Page Footing
HE text [%]	Page Heading
IF filename	Include File
IN [+ -]n	Indent
LM [+ -]n	Left Margin
LS n	Line Spacing
ML filename	Mailing List File
NA	No Right Margin Justification
NF	No Text Filling
PA [+ -] n	Set Page Number
PC n1,n2,...	Printer Control
PL [+ -] n	Page Length
RM [+ -] n	Right Margin
SP n	Space n Lines
TL n:n1,n2,...	Transliterate Character

## SAVE/RECALL, Notes on usage

by Todd Kaplan, USA

The purpose of the above utility is to allow assembly language subprograms to load in a lot faster than the usual way of CALL LOADs with the slow BASIC Loader. This also has the beneficial side-effect of taking up less room on disk, and allowing both Extended BASIC and assembler language programs to reside in the same program.

This utility as is, will only work with relocatable assembler language programs. The reason for this restriction is that absolute code does not change any memory pointer, thus making it kind of hard for the utility to locate where the program is in memory. This utility also saves only the amount of memory that is necessary, that is, none of the utilities like VMBW are saved, because Extended BASIC loads those in when you type in CALL INIT.

## Instructions for Use

Part 1: Saving the assembler language code with your Extended BASIC program.

- (1) Enter Extended BASIC.
- (2) Type NEW
- (3) Type CALL INIT
- (4) Load in ALSAVE by typing  
CALL LOAD("DSK1.ALSAVE")  
(or any appropriate drive).
- (5) After ALSAVE is loaded, load in your assembler language routines in the same manner, for example,  
CALL LOAD("DSK1.A/L-CODE1")  
CALL LOAD("DSK1.A/L-CODES2")  
or whatever file name may be appropriate in your case.
- (6) When all assembler language routines are loaded in, type  
CALL LINK("SAVE")  
This will move the assembler language routines from low memory (Extended BASIC assembler language space) to high memory (Extended BASIC program space) and adjust the necessary pointers to tell Extended BASIC so that the routines will not be over written.
- (7) Type in  
100 REM  
as the first statement.
- (8) Type in  
MERGE DSK1.ALLOADM
- (9) Finally save the program with  
SAVE DSK1.PROGRAM  
or whatever file name you want for the resultant Extended BASIC assembler language program.

Part 2: Using the program created with XBALSAVE.

After the above program is saved, you may add another program after it only by loading in the above program and MERGEing another Extended BASIC program with it. The entire combination can be saved.

The contents of the ALLOADM program are as follows:

```
10 CALL INIT :: CALL LOAD(8196,63,248) ::
CALL LOAD(16376,65,32,32,32,32,32,255,48)::
CALL LINK("A")
```

Here is what each part does:

CALL INIT - Initializes the Extended BASIC assembler language pointers and loads in utilities.

CALL LOAD(8196,63,248) - Loads the address >2004 (LFAL) with the value of >3FF8. This tells Extended BASIC that we have a program name in the name table.

CALL LOAD(16376,65,32,...,32,255,48) - This puts the name 'A' into the name table with an entry address of >FF30 (start of recall program) so that we can access the recall program with a CALL LINK("A").

continued on page 26



# Forth to you too! Session 5

Author unknown

When we set up our autobooting system disk I stated that I always include -DUMP. This utility provides a lot more than what I am going to cover here since I want to keep things as simple and understandable as possible. As you may have gathered by now, Forth is a 'stack-oriented' language. There are several 'stacks' but whenever there is reference made to 'the stack' it means the PARAMETER stack. (Parameter = argument.) The stack's main function is for temporary storage of arguments, i.e., numbers. Every time you enter anything from the keyboard (or a word is encountered in a program) Forth first looks for it in the dictionary. If it is not there, it is converted to a number and put on the stack (If found in the dictionary it is executed). By now you know the stack concept: What went on the stack last will be taken off first. Because there is a limit to your computer's memory capacity, there is also a limit as to what the stack can hold. Good programmers make sure that the stack holds only what is needed and don't let 'garbage' accumulate there. If by chance you define words which leave junk on the stack, it will eventually reach its limit and the program will stop with a ?FULL STACK message on your display. Conversely, if a word needs to fetch a parameter from the stack and nothing is there, you'll get a ?EMPTY STACK error message.

Bring on .S (dot-S). It let's you look at the stack content without touching it otherwise, i.e., it neither adds nor removes anything. For example, let's enter 15 and then 22. Now enter .S and see what you get on the display. It should show | 15 22. The | symbolizes the bottom of the stack. In other words, if we use . (dot) then 22 should be printed to the display because it is the top (first-out) item on the stack. Now use .S and see what's left. Only the 15. Another . (dot) will fetch it and if you use . once more Forth will respond with ? EMPTY STACK. (Usually preceeded by a number.) In order to program in Forth you must understand the whys and hows of the stack.

Speaking of stacks, there is another one, though it is never called a stack. It goes by the name of DICTIONARY, but just like nearly everything in Forth it is also a stack. Every time you define a new word it ends up on top of the dictionary (stack). On the bottom reside - you guessed it - the Forth resident words. Our autoboot then piles the words from the load options on top and finally you add your words (or your program's words). Large programs can use up almost all of the memory. Say you have loaded the AAA SUPER-DUPER XY CALCULATOR and there are now 1,500 bytes free. You are through calculating and wish to install the PARAGON XY PLOTTER. You may not be aware of the fact that it takes 4,000 bytes, so as it boots there comes the point where your TI has reached its limit. ?DICTIONARY FULL will be the message to let you know that there is no way you can run the XY PLOTTER with SUPER-DUPER CALCULATOR still in memory. Well, there's always COLD to start over. Not necessary. FORGET is easier and faster. FORGET cccc (as it is stated in the manual) wipes everything out of memory starting with cccc and every word which was added after it.

One way to always know what to FORGET to get rid of a program, but not the autoboot, is to include on the WELCOME screen a do-nothing word. It should be added as the last word just before the R->BASE word. It can be anything you like, most people use their initials to help them remember to FORGET. (How is that for logic?) Every word compiled prior to my : LW ; remains in the dictionary, every word added afterward is dropped by FORGET LW <ENTER>. In the case of very short routines which I may load on top of another one I usually include a : XX ; or similar do-nothing and display a prompt upon exiting to remind me what to forget. In this fashion I leave the underlying program in memory.

RECAP:

The .S word displays the parameter stack's content without adding or removing anything. "|" denotes the bottom of the stack.

FORGET cccc let's you clear from the dictionary entries beginning with cccc and every word added since cccc was compiled.

Placing a do-nothing word on screen 3 makes forgetting easy.

SUGGESTION: Study Chapter 7 of STARTING FORTH.

## From the Bulletin Board

MAIL TO : ALL  
MAIL FROM : SHANE

Thought that I would take this opportunity to extend to you all a very Happy New Year. Great to see that both the Club and the BBS is still in very good hands and doing great. I enjoy reading my TISHUG News Digest every month, though I am unable to attend the meetings due to work commitments. 1990 should prove very interesting for this club, with the continued dedication and support from all of you, and I wish you all the very best.

Kindest Regards  
SHANE ANDERSEN

MAIL TO : ALL  
MAIL FROM : SARA

Help! somehow I have crashed a new disk by the name of "Math Routine Library" which contains -:

FOURIER SERIES  
ORDINARY DIFFERENTIAL EQUATIONS  
FUNCTIONAL ANALYSIS  
BASES, PRIMES AND HYPERBOLICS  
SIMULTANEOUS EQUATIONS AND MATRIX  
INVERSION

I have the instruction booklet with all the goodies but no operational disk!

Has anyone a copy I could copy? Seems a pity to have to contact the USA supplier if I am able to get a copy in Aussie.

REGARDS JOHN RYAN.

MAIL TO : ALL  
MAIL FROM : BITBITE1

BITS AND BITES 1 is the first part of the TI-Writer Manual. Over the next few months will be doing the WHOLE TI-Writer manual including Version 4.4 It has been set up so it can be easy to reformat it for printing out the manual with the TI-Writer version 4.3/4 Formatter. All .PC (printer codes) should work on most printers, .PC 27,103,1 is double height print, .PC 27,103,2 is quad height print, and .PC 27,103,0 is cancelled double/quad print. Bye for now LARRY

In the disk manual for TI-Writer 4.0/4 there is one error, .CF (Chain Files) is wrong should read .CH (Chain Files) Bye for now Larry.

As at 30th November 1989, PRESS is still with bugs and Chris will not guess a time now when it will be released.

Have a MERRY CHRISTMAS AND A HAPPY NEW YEAR.

Have just received a large amount of diskettes from the USA, have not had time to see what ones they are yet. Bye for now LARRY.

## Getting a line on your Program

by Joe Presser, USA

Curious, George? Have you ever wondered how your TI99/4A console stores and reads the BASIC program you type in? Well, I have! It all started when I began to have problems with one of my consoles.

Every time I edited an existing line, other lines got "messed up" or even moved to some other part of the program! Eventually, I had to have the console replaced, but it did not keep me from becoming curious. I was determined to figure out how this could happen. When I finally found the answer, I learned a great deal about how the TI99/4A keeps track of the program we type into it. Below is some information that you can use to get "computer's-eye-view" of the program you have typed in.

### IT TAKES TWO TO TANGO

Any BASIC operating system has to keep track of every line of code you type in. It not only has to keep track of the program data, but it also has to keep track of the line numbers too! The TI BASIC system uses two different areas for this; 1) the Line Number Table and 2) the Program Area.

All this information is stored in the highest memory available to the console. For TI BASIC that will usually be somewhere around address 14228 and for Extended BASIC with memory expansion, somewhere around address -28. This all depends on the number of files open (as in CALL FILES(1)) and whether the disk drives are attached, etc.

### PROGRAM FIRST

The actual lines of program code are stored first. Each line of code is stored in a series of bytes. Most all reserved words like PRINT, INPUT, REM, etc., are stored as a single byte value called a token. For instance, the token values for the above three words are 156(>9C), 146(>92) and 154(>9A). TIBASIC also keeps track of variables, constants and jump references in a similar way. Each line is preceded by a length byte and, at the end of each line, the value 0(zero) is placed to indicate the end of the line.

As each line is typed in, the BASIC operating system translates it into these byte values and stores it at the top end of the available memory. When you edit a line or insert a new line between existing ones, that new line is stored at the "end" of the Program Area, not inserted into the middle. This is important to remember when trying to locate a line of code! Note also that the line numbers (e.g. 100, 110, etc.) are not stored with the line of code. This allows the TI99/4A system to easily RESequence your program, and allows the system to place lines at any available memory location instead of in execution order only. So where are the line numbers? In the Line Number table, of course!

### THE LINE NUMBER TABLE

After all the lines of code are stored, the Line Number Table begins. Each line of code has 4 (four) bytes of information in the Line Number Table: two bytes for the line number (>0064=line 1000, and two bytes for the memory address where the line of code is stored (>FFE7=-28). When the BASIC interpreter is running the program it looks at the Line Number Table, gets the line number and the line address, branches to the address, gets the actual program line and then performs the line of code! Not so simple, eh?.

Since the Line Number Table is stored at the end of all the program lines, each time you add a new program line, the entire table has to be "shifted" downward into lower memory. This is why, especially in a long TI BASIC program, it takes quite a while for the cursor to "come back" after you have edited a line. The operating

system must code the line, shift all the line numbers and addresses downward and then insert the new line number and address into the table. That would take anybody a little time!

### PEEKING AROUND

Let us look in on the operating system as it does all this stuff! Here is what we will do:

- 1) Enter Extended BASIC (or TI BASIC with Editor Assembler or Mini Memory module) and type:

```
100 PRINT "THIS IS A TEST"
```

- 2) Find Line Number Table. To find the Line Number Table, PEEK into the CPU RAM PAD at address -31950; get two consecutive bytes; convert these bytes to decimal; subtract 3 bytes (overhead) and, if you are using Extended BASIC with memory expansion, subtract 65536; you now know the address where the line table starts!

```
CALL PEEK(-31950,A1,A2)
LT=A1*256+A2-3-65536
```

In Extended BASIC, the value should be -47 and in TI BASIC it should be 14273. As each line of code is typed in, the table will move further down into memory.

- 3) Get the line and address. Using the start of the Line Number Table, get the first line of code and the address where it is stored. In Extended BASIC this is stored in memory expansion so you use the following:

```
CALL PEEK(LT,L1,L2,A1,A2)
LN=L1*256+L2
LA=A1*256+A2-65536
```

```
LT=Line Table Address (-47)
LN=Line Number (100)
LA=Line Address (-42)
```

In TI BASIC, the program is stored in the console itself, therefore you must look into the VDP area with the command PEEKV:

```
CALL PEEKV(LT,L1,L2,A1,A2)
LN=L1*256+L2
LA=A1*256+A2
```

```
LT=Line Table Address (14273)
LN=Line Number (100)
LA=Line Address (14278)
```

- 4) Get the line length. Each line of program code is preceded by a line length byte. This tells the system how many bytes to interpret. To find the line length, read the byte preceding the Line Address:

```
Extended BASIC:
CALL PEEK(LA-1,LL)
```

```
TI BASIC:
CALL PEEKV(LA-1,LL)
```

In both cases, LL=18.

- 5) Read the Program line. Now all we have to do is read in 18 bytes of code starting at the Line Address (LA):

```
Extended BASIC:
CALL PEEK(LA,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R)
PRINT A;B;C;D;E;F;G;H;I;J;K;L;M;N;O;P;Q;R
```

```
TI BASIC:
CALL PEEKV(LA,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R)
PRINT A;B;C;D;E;F;G;H;I;J;K;L;M;N;O;P;Q;R
```

In both cases, the values A to R will equal:

```
156 199 14 84 72 73 83 32 73 83 32 65 32 84 69 83 84 0
```

156 - Token value for PRINT  
 199 - indicates a string value  
 14 - length of string

The rest of the bytes are ASCII values for the string (84=T, 72=H, 73=I, 83=S, etc.).

#### POKING AROUND

Now let us see if we cannot reverse the process. This time we will do a few POKES into memory that will make the computer think you typed in a line of code! Here is what we will do:

- 1) Clear memory and type CALL INIT.
- 2) Set up Line Table. The system needs to know where the line table starts, so:

```
CALL LOAD(-31950,A1,A2)
```

Extended BASIC:  
 A1=255, A2=220

TI BASIC:  
 A1=55, A2=204

- 3) Insert line # and address: Next we need to fill up the line table with line and address data:

Extended BASIC:  
 CALL LOAD(-39,0,100,255,222)

TI BASIC:  
 CALL POKEV(14281,0,100,55,206)

- 4) Insert line length and line code. Now all we need to do is insert the line data:

Extended BASIC:  
 CALL LOAD(-35,10,156,199,6,72,69,76,76,79,33,0)

TI BASIC:  
 CALL PEEKV(14285,10,156,199,6,72,69,76,76,79,33,0)

- 5) List your program! Because the system has not been properly informed as to where the program ends, this is not "runable" code. But at least you can see your work!

#### EXAMPLE PROGRAM

Below is a program that will "read itself". Using the tools we learned above, this routine will read the Line Number Table, and find the Line Address and Line Length. After each line is found, the program waits for you to hit a key before it reads the next line.

Note: This program is written for TI BASIC with Editor Assembler or Mini Memory modules. To use it with the Extended BASIC and memory expansion you need to make a few changes:

- 1) Line 220 - add -65536 to the end of the line.
- 2) Line 260 - change PEEKV to PEEK.
- 3) Line 290 - add -65536 to the end of the line.
- 4) Line 330 - change PEEKV to PEEK.

#### ONWARDS!

Armed with this information, you can begin to learn more about how the TI99/4A system works and you can use these tools to help read and write your own programs. A number of programmers place routines like this into their programs as a way of "protecting" valuable code sequences or algorithms from prying eyes. It is usually called "imbedded code". You can also use routines like this to translate a text line into program format once you learn to decode all the token values!

Next time, we will talk about how to write a routine that writes its own RUNable program!

```
100 REM *****
110 REM *
120 REM * PROGRAM PEEKER *
130 REM *
140 REM *****
150 REM
160 REM      11/84
170 REM      SUBFILE99
180 REM
190 CALL INIT
200 CALL CLEAR
210 REM
220 REM *FIND LINE TABLE*
230 REM
240 CALL PEEK(-31950,A1,A2)
250 LT=A1*256+A2-3
260 REM
270 REM *GET LINE AND ADDRESS *
280 REM
290 CALL PEEKV(LT,L1,L1,A1,A2)
300 LN=L1*256+L2
310 IF LN=0 THEN 570
320 LA=A1*256+A2
330 REM
340 REM *GET LINE LENGTH*
350 REM
360 CALL PEEKV(LA-1,LL)
370 REM
380 REM *****
390 REM *PRINT THE DATA*
400 REM *****
410 REM
420 PRINT "LINE";LN
430 PRINT "ADDR";LA
440 PRINT "LEN ";LL
450 PRINT
460 CALL SOUND(150,1400,0)
470 CALL KEY(O,K,S)
480 IF S=0 THEN 470
490 REM
500 REM *UPDATE LT*
510 REM
520 LT=LT-4
530 GOTO 290
540 REM
550 REM *END*
560 REM
570 PRINT "END OF PROGRAM"
580 PRINT
590 END
```

continued from page 23

CALL LINK("A") - Executes the recall program. The recall program (CALL LINK("A")) does what the save program does, except in reverse. It moves the assembler language program from high memory (where it was saved) back to its original position so that it can be run.

After the CALL LINK("A"), all of the other CALL LINKS that were accessible at the time they were saved are now fully accessible again. They work exactly as if they were just loaded with a CALL LOAD, except you do not have to wait for the slow Extended BASIC loader.

#### Some additional notes

- (1) As SAVE moves data into high memory, this also limits the size of your Extended BASIC program. To regain the original amount of programming space, you must either type NEW or RUN another Extended BASIC program. The above utility could just be used by itself solely as a load program, and then RUN the main program.
- (2) As mentioned above, this will only work with relocatable code. The code also has to reside in the normal assembler language space for use with Extended BASIC, namely >2000 to >3FFF. If the code is relocatable, you will not be able to load it outside of that range anyway!

continued on page 29

# Page Pro 99

by Ed Johnson, USA

## INTRODUCTION

Page Pro 99 has a long and varied history. The program was begun almost two years ago by Chris Bobbitt and Ed Johnson. In its original conception it was simply a modified version of TI-Writer that would allow you to generate a form on the screen with a limited number of line patterns. Originally, we decided to call the program "Form Maker 99" (you may have seen it advertised under this name). The program worked fine, but we were not satisfied.

The program was not capable of doing things that form programs for other computers could do. So Ed and Chris thought of a lot of things they both wanted to see in the program; in retrospect things which re-defined the concept they originally had. Ed started from scratch in assembly and what you see before you is the result of nearly two years of development and "fine tuning". Page Pro 99 is a program that, to put it simply, allows you to create full page forms. With Page Pro 99 you can type in any of four different directions using a small or large text font, draw lines with line font, and you can include up to 28 pictures of any size anywhere on the page. You can "import" a TI-Writer text file and "export" a page as a text file for use in other programs. You can even print out the page in different dot densities (ranging from a "rough draft" single density to reproduction quality quadruple density).

Due to the vast changes in the program over its course of development, the name "Form Maker 99" just did not seem to do justice to its capabilities. At the recommendation of a very good friend (thanks Denny), it was decided to change the name to "Page Pro 99". We hope that you will agree that the performance of this program lives up to its new name.

Page Pro 99 will, in effect, let you do much of what a dedicated desk top publisher will do, much simpler and faster. Page Pro 99 is the only program of its type that is entirely "what you see is what you get". Whatever text, lines, or pictures you place on the screen will appear exactly as they will on your paper. Unlike other desk top publishing programs, it is easy to create impressive looking letter heads, signs, charts, maps, graphs, and even forms.

However, we are not going to call Page Pro 99 a true "desk top publisher". Why? Because Page Pro 99 will only let you have three different fonts on the page at once and only 28 pictures. It really is not fair to call it a desk top publisher, even though it does more in this area than most other programs that call themselves that. We simply do not want to raise everyone's expectations of what it will do. We will not object if you would like to use Page Pro 99 as a desk top publisher, or even if you would like to call it one. We simply will not do so. We use it to make more than just forms, and think that you will too!

## LOADING INSTRUCTIONS

Before doing anything else, make a backup copy of the program and use the backup copy.

### Editor Assembler Module

Select E/A option #5 - Load Program File. Press <ENTER> for the default name "UTIL1" (even though no filename appears. This loader will look for DSK1.UTIL1). The program will then load and run.

### TI-Writer Module

Select option #3 - Utilities. Press <ENTER> for the default name that appears on the screen (DSK1.UTIL1). The program will then load and run.

## Extended BASIC Module

With the program disk in drive #1, the program will load and run automatically after selecting Extended Basic from the menu screen.

## RAMdisks

Will load from any RAMdisks, loading with UTIL1. Note program runs best when using the files, pictures, etc from a RAMdisk.

## PROGRAM CONCEPTS

All of the functions of Page Pro 99 are in one program. When the program first loads and runs, a title screen will appear in the upper 3/4 of the screen. This upper 3/4 area is the "page window" to a much larger area, which is the entire page.

What you see is 12 lines by 31 columns of the total 66 lines by 60 columns of the page. As you move around the page, the window will adjust itself to display the part of the page you are on. The only way to view the whole page is to print it out. The 66 line by 60 column page of Page Pro 99 is almost the same size as a standard typewritten page (a standard text page is 66 by 80). When printed out, it will fill 8 1/2 by 11 inch page from perforation to perforation, with a small margin on the left and right. Proper paper alignment is very important.

The bottom 1/4 of the screen is used for displaying the "status" line, prompts, and for getting the keyboard input for filenames, etc. The status line keeps you up to date on the current cursor position on the page, the direction the cursor will move as you type, and whether or not the picture display is "ON or OFF" (more on this later!).

## GETTING STARTED

Once you understand how the program represents a page you can actually get started in creating your own layout for a page. Once you are familiar with the cursor controls and how they work, you can easily begin typing in your text.

Note that as you type, when reaching the end of a line the cursor automatically wraps down to the next line. However, any word you may have been typing at the right hand side of the page will be split in two. Page Pro 99 does not support "word wrap"; in other words, it will not automatically keep words connected between lines. If this editing limitation is cumbersome to you Page Pro 99 will let you use any text editor that saves files in Display Variable 80 format to generate the bulk of your text, and then you can "import" it into Pro Page 99 to put the "finishing touches" to it.

## CURSOR CONTROL KEYS

Arrow keys - FCTN[E], [S], [D] and [X]

These keys will simply move around your page without disturbing whatever you have on a page.

Page Down - FCTN[4]

Move the cursor down the page 12 lines at a time until it reaches the last line (line 66).

Page up - FCTN[6]

Moves the cursor up the page 12 lines at a time until it reaches the first line (line 1).

Window left or right - FCTN[5]

Moves the cursor forward or back 30 columns to the window opposite of the one you are currently in.

## CURSOR AUTO - MOVEMENT

Direction - CTRL[E], [S], [D] and [X]

These keys control which direction the cursor will move after pressing a key when editing your text or drawing lines. The arrow in the lower right hand corner

of the screen indicates the direction you will move as you type. Page Pro 99 will always treat the cursor movement as a left to right in terms of where the cursor will end up after pressing <ENTER> key or when it wraps to the next line. That is, if you are typing "backwards" (right to left), the cursor will move up one line after <ENTER> is pressed or if you "WRAP" left of column #1. Similarly, if you are typing down the page, you will end up the top of the page one column to the left after pressing <ENTER> or if you wrap off the bottom of the page.

This may seem trivial or too confusing but it will make entering text with a sideways or upside down font much easier (anyone thinking "greeting cards" yet? We were!). Spend some time getting used to typing in different directions. You will soon be a real "pro" getting around with Page Pro 99!

#### EDITING KEYS

FCTN[E] - Move cursor up  
 FCTN[X] - Move cursor down  
 FCTN[S] - Move cursor left  
 FCTN[D] - Move cursor right  
 FCTN[1] - Delete character  
 FCTN[2] - Insert character  
 FCTN[3] - Delete line  
 FCTN[4] - Down one screen  
 FCTN[5] - Next screen  
 FCTN[6] - Up one screen  
 FCTN[8] - Insert line

#### Delete character - FCTN[1]

Is only active when typing small text or line characters. It will delete a single character and pull all characters to the right of the cursor one position to the left. Large text and pictures are not shifted, nor are any characters to the right of a large text character or picture affected by the delete.

#### Insert character - FCTN[2]

Is only active when typing small text or line characters. It will insert a single space and push all characters from the cursor to the right one position. Large characters as well as pictures are not affected by any insertion, nor are any characters to the right of a large text or pictures. Insert does not stay "ON" it just inserts a single space.

#### Insert line - FCTN[8]

Is active at all times and will insert a blank line at the current line the cursor is on. Some care must be taken to not "cut in half" any large text or pictures. This may result in some unwanted results or cause an error when the program tries to load a picture from disk. (Handling this type of situation is discussed in more detail in the "Tips and Tricks" section of this manual.) All lines from the current line down are pushed down to make room for the new line. The last line of text (if line 66) is pushed off the bottom of the page and can not be recovered.

#### DRAWING LINES

At some point you will want to draw some lines on your page. You may need a line to draw a table, a decoration, a border for a sign or memo, or to delineate different parts of the page. In Page Pro 99, a line character is actually part of a "FONT". There are 30 line graphics representing straight pieces, corners, cross pieces, T pieces and other connectors. These line graphics are attached to the keys A through to O (upper and lower case) on the keyboard. The first 15 line graphics are placed by using lower case a to o.

To get into the line graphics mode, press CTRL[8]. This key acts like a switch to turn the line mode on and off. To leave this mode, Just press CTRL[8] again. Note that the cursor changes when you turn this mode on. A small "L" appears in the center of the cursor to indicate that you are now in the line graphics mode. Much easier to notice is the line "Reference" that appears in the status area whenever the line graphics

mode is on. This reference is there to help remind you which characters are attached to which keys. If this display is a bit confusing at first, you may want to load and print the sample page "LINE-AID" that was included with the program and use it as a guide until you have become familiar with the ideas.

To load a picture and place it on your page, position the cursor in the upper left hand corner of where you want the picture to start, and press CTRL[L]. Page Pro 99 will ask you for the path and filename of the picture you want to load. The picture file is read to determine its size and find all the "white" spaces. If the picture display is turned on, you will then see as much of the picture as will fit on the screen. If the picture display is turned off, you will see a bunch of inverse "P"'s on the screen in the place of your picture. Note: the picture can be a font like the examples on this page (Loading) and (pictures). CTRL[0] (oh, not zero) acts as a toggle switch for turning the picture display on and off. When the program first starts, picture display is on (indicated by the inverse P in the centre of the line/column status line). To turn the picture display off, press CTRL[0]. You will see the P in the status line change to a "\*" to indicate that the picture display mode is off. This feature is included mainly for speed purposes. Page Pro 99 does not hold the picture in memory, it just keeps track of where it is located on the page and where the picture can be found on your disk. All pictures that have been loaded must remain in your disk drive(s) so Page Pro can find them! It is highly recommended that you copy any picture files you plan on using to your work disk before starting a session.

There are 3 ways to delete pictures from a page. The first is to simply type text over the area the picture occupies. You can type text over any part of a picture or the whole thing if you wish. There are advantages to typing over parts of a picture, but there is little advantage in using this method to erase an entire picture. If you use this method to delete a picture, the picture data will not be deleted from the table in the program that keeps track of the pictures. This is not a big problem unless you plan on using all the 28 pictures that are available. The second way to delete a picture, and this is the recommended way, is to move the cursor onto any part or portion of the picture you want to delete and press CTRL[K] (if you think of this operation as "killing" the picture it will be easy to remember CTRL[K]). The status area will display a prompt asking you if you indeed want to delete the picture. Press "Y" to confirm your intent. If the display prompt tells you that a picture is not located where the cursor is, it means that you are not on the picture or are located on some "white" space in the middle of the picture area (the easiest way to see where the "white" spaces are located is to turn the picture display off). Just re-position the cursor and try again.

The third way to delete a picture is pressing CTRL[U]. This operation will delete all pictures on your page and reset the table that keeps track of picture locations. Use caution with operation. Once you confirm your action, all pictures will be erased.

Once you have created a page, you may want to save it for future editing or printing. Page Pro 99 provides 2 ways for you to save your work. Press CTRL[F] to bring up the file option menu. To save your page along with your picture information, select option 2: Save Page. This selection will save your page into a 21 sector "memory image" file that contains your text as well as your picture data. Option #4: (Export Text) should be used when you just want to save the text for use in other programs. The picture data will not be saved with this option.

To load a previously saved page, just bring up the file options menu by pressing CTRL[F]. Select option #1: (Load Page) to load a page that was previously saved with with option #2. To load a text file that was saved with option #4 or created using another program (such as

TI-Writer), select option #3: Import Text. This option will read any D/V80 file and load it into Page Pro 99. Note that the limit is 66 lines by 60 columns of text. Any text beyond 60 characters per line or beyond 66 lines will be ignored. Also, remember that when you load a page (option #1) the picture data is also loaded. Pictures should be in the same drives that they were in when the page was saved. See the "Tips and Tricks" section for some ideas on handling this.

Page Pro 99 can hold 3 different fonts (sometimes 4) in memory at the same time: a small text font, a large text font, and a line graphics font. Each of these fonts can be changed by loading in a new font from disk. To load an alternate font, press CTRL[A]. A sub-menu menu will be displayed offering you a choice of which new font to load. Press "1" to load a new line graphic font, "2" to load a new small text font, or "3" to load a new large text font. Next, you will be asked to enter the path and filename of the new font. The new font will then be loaded and replace the old font in memory. Page Pro 99 can only support one of each of the three different fonts at a time. (Upper case and lower case in text fonts, so it is possible to fit 2 fonts into a large or small font if one of the fonts uses upper case only and is loaded in last.) With creative use of the picture capabilities it is easy to give the appearance of more fonts on a page. See the "Tips and Tricks" section for some ideas.

Printing a page is simple with Page Pro 99. Just press CTRL[P] and you will be prompted to enter the path and file name of the device you want to send your output to (it is possible to "print" a page to disk, but be warned to the resulting file will be very large! A single-density print file on disk will take up around 200 sectors, a double-density about 400 and quad-density about 800.

After entering the device name for your printer, you will be prompted to select the density of your printer output. Most of the newer Epson compatible printers will support all three of the density selection offered (single, double, and quadruple), but many of the older models will only support single and double. Check your printer manual if you are not sure. Press 1, 2 or 3 to select the desired print density. Next, the program will check to see if the picture display mode is turned on. If not, you will be asked if you want to turn the pictures on. Press "Y" to turn pictures on or any other key to leave them off. If you print a page with pictures turned off, the inverse "p"s you see displayed on the screen will be printed instead of your pictures. Why would you want to do this? If you want a "rough draft" to check the layout of a page, it is a bit faster to get a print out with the pictures turned off since they do not have to be read in from disk.

Everything is now set to print. You will be reminded to make sure your printer is turned on and ready to go (paper aligned, etc.). Press <ENTER> to start the printing process. The screen will blank while the page is printed. This is a necessary evil and is nothing to be concerned about.

If you need to abort printing for any reason, just press FCTN[4]. If you are using Page Pro 99 with print spooler, it may be necessary to turn your printer off to totally stop the printer.

Press CTRL[C] to change the screen colours of Page Pro 99. There are 12 different colour combinations in all that should cover most preferences for various types of display monitors.

At virtually all prompts when Page Pro 99 is asking for information (filenames, menu selection, etc.), you can cancel the operation by pressing FCTN[9]. If this is done when entering a filename, the name will be erased and the next time you access that particular mode, you will have to re-enter the filename (the default name is also erased).

continued on page 30

continued from page 26

(3) The above is almost exactly like my old XBALSAVE, except the method of running the RECALL program has been changed to make it more reliable. It used to load the Interrupt Service Routine (ISR) hook at >83C4 with the starting address of the RECALL program, but that only worked at random.

\*\*\*\*\*

\* SAVE/RECALL \* Source Code

\*\*\*\*\*

\* By Todd Kaplan

\*

\* Note, User may wish to change the name of 'SAVE' as

\* it might be one of the user's assembly CALL LINKS.

```

DEF SAVE
M24F4 EQU >24F4
START EQU >FF30      Start of Recall Program
AORG START
* start of RECALL program
LIMI 0               disable interrupts
MOV R11,@SAVRTN     save return address
LWPI MYREG           get workspace
* restore utility data
LI R0,M2000          Get Address of Storage Block
LI R1,>2000           Get Place to Move To
LI R2,4              4 Words to Copy
BL @MOVE             Move it!
* restore REF/DEF table
MOV @RSTART,R0       get start of REF/DEF buffer
MOV @>2004,R1         get start of REF/DEF table
MOV @RLEN,R2          get word length
BL @MOVE             move it
* RESTORE program
MOV @PSTART,R0       get start of program buffer
LI R1,M24F4          get start of program
MOV @PLEN,R2          get program word count
BL @MOVE             move the junk
* return to Extended BASIC
RETURN
SB @>837C,@>837C     clear GPL status byte
LWPI >83E0            register workspace
MOV @SAVRTN,R11      restore return address
RT                   return to calling program
*****
* MOVE * Move CPU TO CPU
*****
* R0 = From: Address *must be even address*
* R1 = To: Address *must be even address*
* R2 = # of words to move
* BL @MOVE
MOVE
MOV *R0+,*R1+ Move a word from *R0 to *R1
DEC R2              Decrement Pointer
JNE MOVE
RT                  Return
* storage
M2000 BSS 2        XML vector
M2002 BSS 2        First Free Address in low mem
M2004 BSS 2        Last Free Address in low mem (start of
                    REF/DEF table)
M2006 DATA >AA55 validation code
RLEN BSS 2         storage for REF/DEF table length
RSTART BSS 2       start of buffer for REF/DEF table
PLEN BSS 2         storage for program length
PSTART BSS 2       start of buffer for program
* end of recall program
*****
* SAVE * save it!
*****
AORG >A010 put in >A000 so this is not saved with
                    recall!
MYREG BSS 32 Work Space
SAVRTN BSS 2 Return Address Storage
SAVE
LIMI 0             clear interrupts
MOV R11,@SAVRTN   save return address
LWPI MYREG         get workspace
* save utility pointers (>2000->2004)
LI R0,>2000
LI R1,M2000
LI R2,3            3 words
BL @MOVE

```

continued on page 31



continued from page 1

I have also been busy with repairs which have been accumulating as my free time has been so scarce. We had a visit from Cedric Que Noy from Darwin with a console and mini-PE system which were not working together. The problem seemed to be with bad contacts at the Cartridge port, the I/O port and possible bad solder joints on the disk controller board. Putting it all together and all seems OK, but it is difficult to be absolutely sure for someone who lives so far away. I have also had a console from Robert Kelly of Babinda and a Corcomp mini expansion system from John Hagart of Gordonvale both from North Queensland. Fortunately, both were easily fixed but once again it is a problem for people so far away. I fixed one of John Hagart's consoles last year and it ran for about three hours after I returned it and then stopped again. When he mailed it down the second time, I then swapped the motherboard and have not yet found what the second problem was. That meant delays and cost of transport so I try and run repaired items for a few hours before I send them back as fixed. I hope to have all repairs done by the February meeting, but some do not look very easy at the moment. Well that is when I start to learn things so I will just have to put in the time.

\*\*\*\*\*

Speaking of the February meeting, I hope that you will all make sure that you come along and support the directors. You could even volunteer for some of the jobs that Dick feels need doing (hoping that Dick will keep going!). Jobs in a group such as ours do not need experts but they need enthusiasts and people willing to take the time to do a job to the best of their ability. The future still looks good for a few more years but we shall have to watch our expenditures very carefully if we are not to get into a shortage of funds situation. We may also need to look at moving out of the incorporation umbrella to save money and perhaps someone could investigate the pros and cons of such an idea. As far as I am concerned, there was not much of an informed debate when we did incorporate and some of the problems were not explained well, if at all. Does anyone know of any benefit that has occurred as a result of incorporation? I can think of a few hundred things we have lost (\$\$).

\*\*\*\*\*

Now to the newsletters received over the break. Firstly the Australian ones starting with ATICC from Adelaide, their November issue. This has a resume of the Melbourne TI-Faire from Fred Cugley and several articles already printed before including two on joystick adaptors.

Bug Bytes of November from Brisbane has a review of the program BOOT providing menus for systems without a RAMdisk, a report on the illegal release of HardMaster on a BBS in the USA, adjusting the modulator to reduce the buzzing noise and an answer to Garry Christensen's puzzle about the Towers of Hanoi problem.

Hunter Valley Newsletters of October and November contain an article on CALL KEY, Tony McGovern looks inside TI-Writer, Richard Terry talks about Flow diagrams and Forth, a joystick adaptor from Al Lawrence, a review of Forthprint by Richard Terry and a list of BASIC tokens.

From Canada we had CIM99 of October and November from Montreal and so mainly in French. There are hints for Zork III and articles on the TI-Expo in Washington, source file for CHARAL and hints for Ghost Town.

Ottawa Newsletters for September, October and November contain articles on Music-Pro by David Caron, a review of Hypercopy and EXEC for the Geneve, assembler language with R.A. Green, Fast Extended BASIC with Lucie Dorais, more on piracy of HardMaster, assembly utility programs for Extended BASIC from David Caron and a review of Chainlink.

TI Focus for November from Hamilton mentions the new addition to the Horizon RAMdisk, RAMBO, creating banners with Giant Artist Posters by Tom Arnold, a review of Editor Assembler version 6.1c and a review of the Chicago TI Show by Tom Arnold.

Finally, from the USA comes Northern NJ newsletters of November with a review of the Dijit AVPC card from Charles Good (of Lima group), another review on the Rave

99 memory enhancement system from Jim McLaren and a way to avoid static damage from Dave Sontos.

The Pug Peripheral from Pittsburgh of September and November have articles on High resolution graphics by Anna Dhein, console repairs help from John Guion, Forth tip #4 from Lutz Winkler, review of Bridge tutorial and Rubber Bridge from Joe Simmons, printer problems from John Willforth, order form for a Zenoboard and notes on the Zenoboard.

ROM from Orange County of November has page 4 missing and mainly has their club news in the rest.

Spirit of 99 for November and December from Central Ohio has a bug fix for R.A. Green's TI-Writer version 4.3 formatter from the author, programming 4-dimensional graphics from Jim Peterson, a review of TI-Artist Plus from Ken Marshall and Dick Beery, TI-Base Tutorial from Martin Smoley and reprints of articles from the TND by Ross Mudie and Lou Amadio.

TIC TOC of October and November from Rocky Mountains Group has a copy of an article about the "TI-99/4A lives on!" from a professional journal, loading programs from disk and lists of programs with useful information about them.

Tidbits of November from Memphis has a review of the Chicago faire, Four-a/Talk from Bill Gaskill and a review of the Milwaukee fair.

\*\*\*\*\*

o

continued from page 29

Under certain circumstances it is necessary to turn the screen display off briefly (when inserting and deleting lines, saving or loading a page, and during printing). Do not be alarmed by this. It is intentionally done to save you from viewing some "garbage" for a moment while the operation is taking place.

Whenever an "I/O error" occurs, the picture display mode is automatically turned off. This is done so that you do not necessarily have to always have your pictures located in the exact same disk drive(s) as when a page was saved. If loading a picture did not cause the error, simply press CTRL[O] to turn them back on. (Any I/O error will cause the cursor to reset to the top left hand corner of the page.)

Page Pro 99 comes with a collection of utilities designed to expand your page making capabilities. These utilities allow you to convert standard artwork into Page Pro's format, as well as manipulate text files for use in Page Pro 99. Future utility packages will enhance the capabilities of the program in remarkable new directions.

All of Page Pro Utilities are available from the Main Menu from Extended BASIC. Follow the Extended BASIC procedure as outline in the first part of this Manual. When the program menu appeared, select the Utility desired by pressing the key of its number. When the Utility has finished executing, it will return you to this Main Menu, where you can load and run other utilities or the program itself, or quit.

The Columnizer program is selected by pressing the key of the number to the left of its name on the Main Menu.

Please note before continuing, this utility assumes (a) some experience with TI-Writer, and (b), some experience with the TI99/4A in general. If you do not have both, you may want to skip this section as it could easily be an exercise in frustration.

This utility is very simple in concept, but can be tricky in execution. In a nutshell: the program takes any TI-Writer text file, converts it into 22 to 29 character columns, and breaks it into files of exactly 1 page each. After it has done this, you load Page Pro 99, and use the Import Text file to load in each page individually so that you can add titles and pictures. Finally, you print each page and/or save it to disk and move on to the next.

o



# Regional Group Reports

## Meeting summary

Banana Coast	11/02/89 Sawtell
Carlingford	21/02/89 Carlingford
Central Coast	10/02/89 Saratoga
Glebe	08/02/89 Glebe
Illawarra	19/02/89 Keiraville
Liverpool	09/02/89
Northern Suburbs	23/02/89 ????
Sutherland	16/02/89 Jannali

### BANANA COAST Regional Group (Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month at 2 pm sharp. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Bayldon, telephone (066)53 2649, or John Ryan of Mullaway via the BBS, user name SARA, or telephone (066)54 1451.

### CARLINGFORD Regional Group

Regular meetings are normally on the third Wednesday of each month at 7.30pm. Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02)871 7753, for more information.

### CENTRAL COAST Regional Group

Regular meetings are now normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043)69 3990. Contact Russell Welham (043)92 4000.

### GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce St, Glebe. Contact Mike Slattery, (02)692 0559.

### ILLAWARRA Regional Group

Regular meetings are normally on the third Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. Contact Lou Amadio on (042)28 4906 for more information.

### LIVERPOOL Regional Group

Regular meeting date is the Friday following the TishUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02)644 7377 (home) or (02)642 7418 (work) for more information.

### NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132.

Come and join in our fun. Dick Warburton.

### SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Group co-ordinator is Peter Young, (02) 528 8775. BBS Contact is Gary Wilson, user name VK2YGW on this BBS.

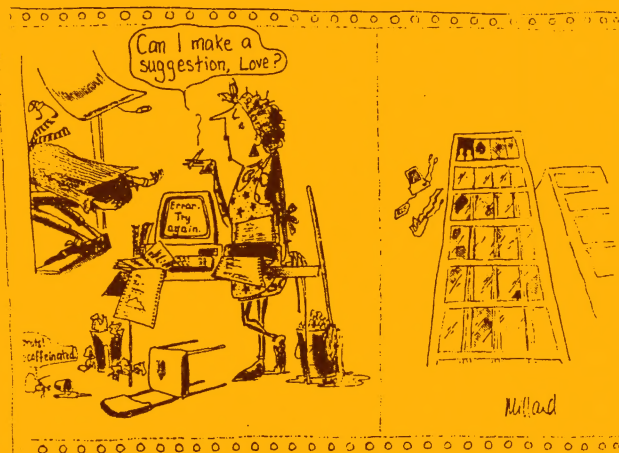
### TiSHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the Woodstock Community Centre, Church street, Burwood. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

Craig Sheehan (Meeting coordinator).

continued from page 29

```
* save REF/DEF table
* figure out length of REF/DEF table (in bytes)
  LI  R2,>4000    get end of REF/DEF table
  MOV @>2004,R0   get start of REF/DEF table
  S    R0,R2      subtract start of REF/DEF table
                  from end to get length
* figure out start of buffer for REF/DEF table
  LI  R1,START    get address of start of this
                  program
  S    R2,R1      subtract length of REF/DEF table
                  to get start of buffer
  MOV  R1,@RSTART store start of REF/DEF table
  SRL  R2,1       divide byte count by 2 to get
                  word count for move
  MOV  R2,@RLEN   store REF/DEF table length (in
                  words)
* copy REF/DEF table to buffer
* R0 = start of REF/DEF table
* R1 = start of buffer for REF/DEF table
* R2 = word length of REF/DEF table
  BL  @MOVE       copy REF/DEF table to CPU buffer
* Save program
* figure out program length
  LI  R0,M24F4    get start of program space
  MOV @>2002,R2    get end of program space (FFAL)
  S    R0,R2      subtract start from end to get
                  byte length
* figure out address of start of buffer for program
  MOV @RSTART,R1  get start of buffer for REF/DEF
                  table
  S    R2,R1      subtract program byte length
                  from start of REF/DEF buff
  MOV  R1,@PSTART store start of program buffer
  SRL  R2,1       divide byte count by two to get
                  word count
  MOV  R2,@PLEN   store program word length
  BL  @MOVE       copy program to program buffer
* set up new bottom of Extended BASIC
  MOV @PSTART,R0  get start of program buffer
  AI  R0,->10     leave a little bit of space
                  between program and Extended
                  BASIC
* set up start and end of Extended BASIC line number
  table
  MOV  R0,@>8330
  MOV  R0,@>8332
  DEC  R0          Back one byte
  MOV  R0,@>8386   store first free address in high
                  memory
  B    @RETURN     return to Extended BASIC
  END
```



## For Sale

Horizon RAMdisk, 192K bytes of RAM in working order, \$260 or \$280 with EPROM. Contact on (044)21 4274 evenings up to 10 pm.